

# Understanding and Applications of AI from Engineering Perspective

SNU  
2024.04.04, 4PM  
홍성걸

X x +

4 4 4 4

—

→

X



**LARGE NATURE MODEL 1**

**LARGE NATURE MODEL 2**

**ABOUT 1**

WITH THE IMMENSE ADDITIONAL DATA CAPTURED, ONE CAN ENGAGE WITH GENERATIVE NATURE SIMULATIONS THAT ARE ANCHORED IN REAL, HUMAN-COLLECTED DATA, INVOLVING KNOWLEDGE IN THE VISUAL SOUND OF A NATURAL REALM TO PRODUCE ART THAT IS ROOTED IN THE CORE OF OUR PLANET'S BEATS.

**ABOUT 2**

THESE SIMULATIONS ARE GENERATED BY AN AI MODEL TRAINED ON A MASSIVE ARCHIVE OF LANDSCAPE IMAGES, CAPTURED BY A NETWORK OF CAMERAS AND DRONES, WHICH ARE USED TO TRAIN A GENERATIVE AI MODEL THAT CAN PRODUCE ART THAT IS ROOTED IN THE CORE OF OUR PLANET'S BEATS.

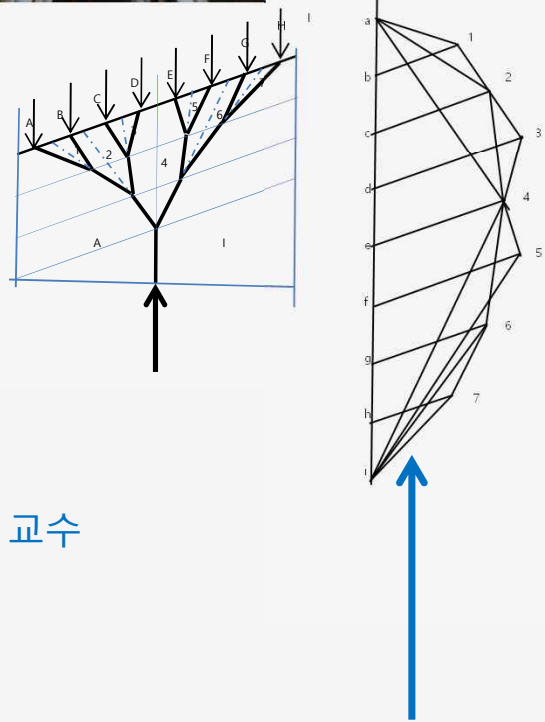
**Echoes of the Earth: Living Archive**

**DATALAND WILL UNITE PIONEERS IN DIVERSE FIELDS INCLUDING THE ARTS, SCIENTIFIC RESEARCHERS, INSTITUTIONAL ARCHIVES, AND CUTTING-EDGE TECHNOLOGY UNDER THE ARTISTIC LEADERSHIP OF REFIK ANADOL STUDIO.**

**LARGE NATURE MODEL 1**

**FRAME PREDICTED HALLUCINATIONS**

Using the algorithm StyleGAN2-ADA (developed by NVIDIA researchers) to capture the machine's "hallucinations" of California landscapes and colors in a multi-dimensional space, Anadol and his team trained a unique AI model with subsets of the collected image archive. Each image in the series displays a cluster of chosen "hallucinations," and Anadol makes selections from an infinite number of images generated by "the machine-mind." The artist explores the latent space of this data universe with a Latent Space Browser - a custom software developed by Refik Anadol Studio in 2017.

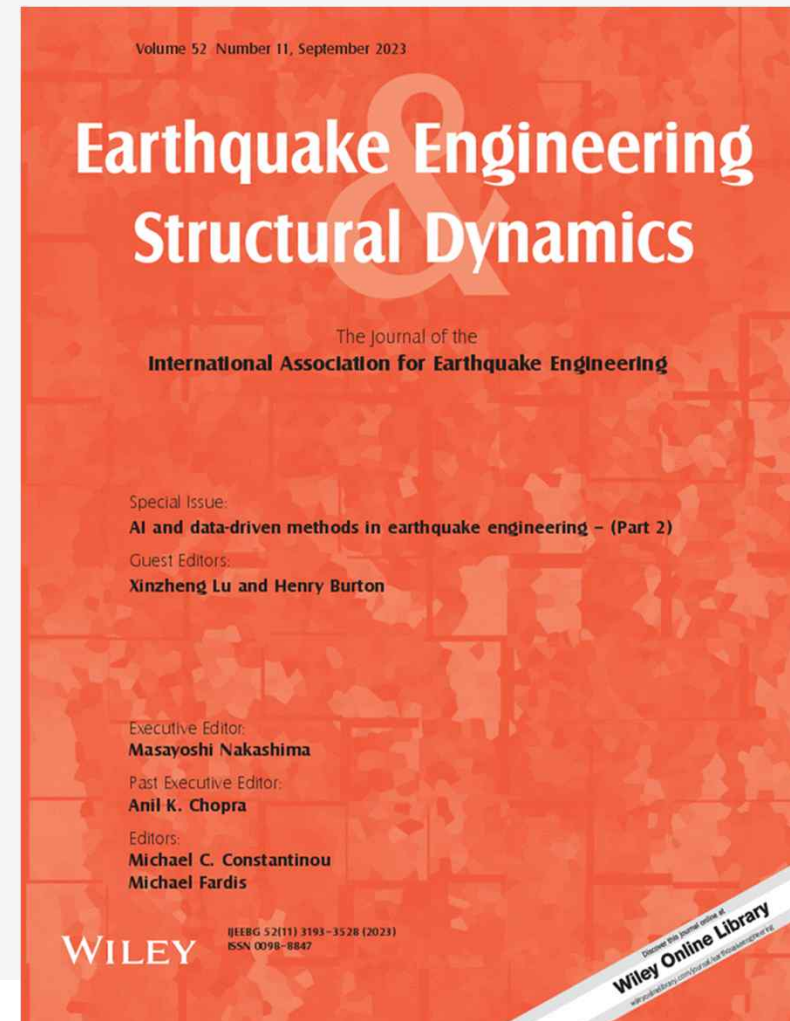
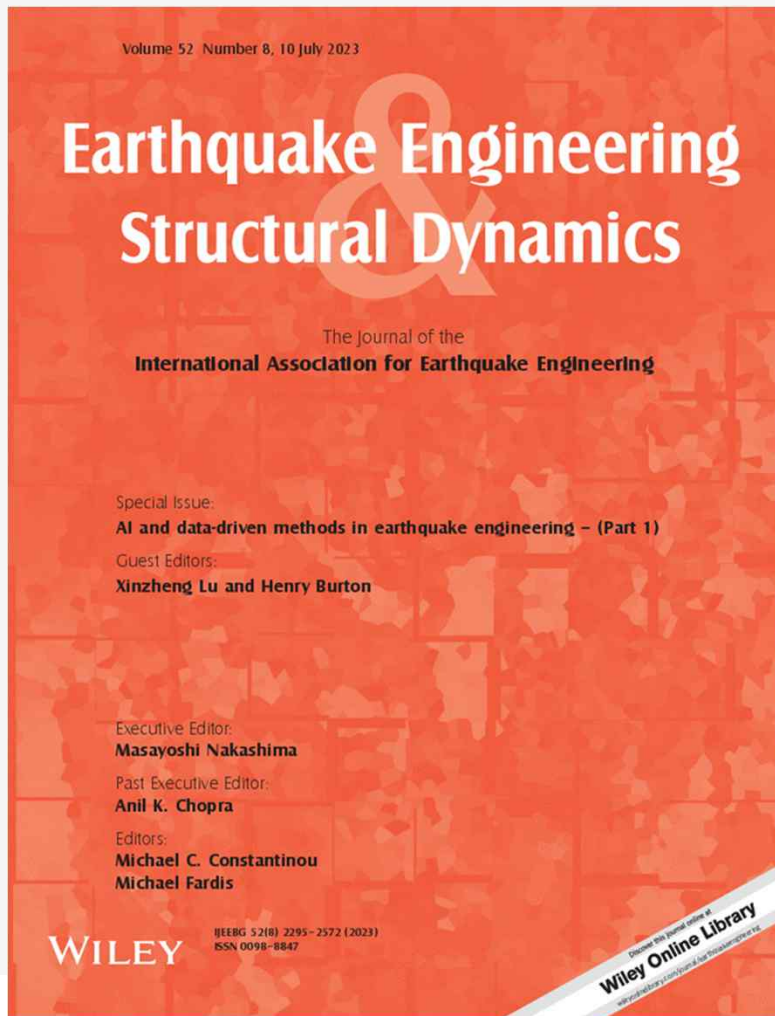


홍성걸 [sglhong@snu.ac.kr](mailto:sglhong@snu.ac.kr)

1996년 이후 서울대 건축학과 교수  
 전 서울대 중앙도서관 관장  
 전 지진공학회 회장  
 전 콘크리트학회 부회장

건축구조시스템과 구조설계, 전통 목조 및 석탑, Graphic statics,  
 콘크리트재료에 관심이 많으며  
 첨단건설재료(UHPC) 개발을 통해 3-D 프린팅으로 구조시스템  
 개발 연구 그리고 AI applications와 Quantum computing

# AI Application to Earthquake Engineering



## SPECIAL ISSUE ARTICLES

Full Access

### [\*\*A hybrid non-parametric ground motion model for shallow crustal earthquakes in Europe\*\*](#)

[Vemula Sreenath](#), [Bhargavi Podili](#), [S. T. G. Raghukanth](#)

### [\*\*Site classification using deep-learning-based image recognition techniques\*\*](#)

[Kun Ji](#), [Chuanbin Zhu](#), [Saman Yaghmaei-Sabegh](#), [Jianqi Lu](#), [Yefei Ren](#), [Ruizhi Wen](#)

### [\*\*Assessment of ground motion amplitude scaling using interpretable Gaussian process regression: Application to steel moment frames\*\*](#)

[Jawad Fayaz](#), [Pablo Torres-Rodas](#), [Miguel Medalla](#), [Farzad Naeim](#)

### [\*\*An unsupervised machine learning based ground motion selection method for computationally efficient estimation of seismic fragility\*\*](#)

[Jinjun Hu](#), [Bali Liu](#), [Lili Xie](#)

### [\*\*Deep learning based seismic response prediction of hysteretic systems having degradation and pinching\*\*](#)

[Taeyong Kim](#), [Oh-Sung Kwon](#), [Junho Song](#)

### [\*\*Surrogate modeling of structural seismic response using probabilistic learning on manifolds\*\*](#)

[Kuanshi Zhong](#), [Javier G. Navarro](#), [Sanjay Govindjee](#), [Gregory G. Deierlein](#)

### [\*\*Combination of physics-based and data-driven modeling for nonlinear structural seismic response prediction through deep residual learning\*\*](#)

[Jia Guo](#), [Ryuta Enokida](#), [Dawei Li](#), [Kohju Ikago](#)

### [\*\*The automated collapse data constructor technique and the data-driven methodology for seismic collapse risk assessment\*\*](#)

[Nenad Bijelić](#), [Dimitrios G. Lignos](#), [Alexandre Alahi](#)

### [\*\*Reliability analysis of structures using probability density evolution method and stochastic spectral embedding surrogate model\*\*](#)

[Sourav Das](#), [Solomon Tesfamariam](#)

### [\*\*Deep learning-based evaluation for mechanical property degradation of seismically damaged RC columns\*\*](#)

[Zenghui Miao](#), [Xiaodong Ji](#), [Minghui Wu](#), [Xiang Gao](#)

### [\*\*Instance segmentation of soft-story buildings from street-view images with semiautomatic annotation\*\*](#)

[Chaofeng Wang](#), [Sascha Hornauer](#), [Stella X. Yu](#), [Frank McKenna](#), [Kincho H. Law](#)

### [\*\*Three-dimensional fragility surface for reinforced concrete shear walls using image-based damage features\*\*](#)

[Amir Hossein Asjodi](#), [Kiarash M. Dolatshahi](#), [Henry V. Burton](#)

### [\*\*Story drift and damage level estimation of buildings using relative acceleration responses with multi-target deep learning models under seismic excitation\*\*](#)

[Jau-Yu Chou](#), [Chieh-Yu Liu](#), [Chia-Ming Chang](#)

## SPECIAL ISSUE ARTICLES

Full Access

### [Unsupervised machine learning for detecting soil layer boundaries from cone penetration test data](#)

[Kenneth S. Hudson](#), [Kristin J. Ulmer](#), [Paolo Zimmaro](#), [Steven L. Kramer](#), [Jonathan P. Stewart](#), [Scott J. Brandenburg](#)

### [A continuous Bayesian network regression model for estimating seismic liquefaction-induced settlement of the free-field ground](#)

[Jilei Hu](#), [Bin Xiong](#), [Zheng Zhang](#), [Jing Wang](#)

### [Machine learning-based prediction of the seismic response of fault-crossing natural gas pipelines](#)

[Wenyang Zhang](#), [Francois Ayello](#), [Doug Honegger](#), [Yousef Bozorgnia](#), [Ertugrul Taciroglu](#)

### [Machine-learning-based optimization framework to support recovery-based design](#)

[Omar Issa](#), [Rodrigo Silva-Lopez](#), [Jack W. Baker](#), [Henry V. Burton](#)

### [Base-isolation design of shear wall structures using physics-rule-co-guided self-supervised generative adversarial networks](#)

[Wenjie Liao](#), [Xinyu Wang](#), [Yifan Fei](#), [Yuli Huang](#), [Linlin Xie](#), [Xinzheng Lu](#)

### [A deep learning method to monitor axial pressure and shear deformation of rubber bearings under coupled compression and shear loading](#)

### [Collaborative filtering-based collapse fragility assessment](#)

[Xingquan Guan](#), [Henry V. Burton](#)

### [Uncertainty-aware structural damage warning system using deep variational composite neural networks](#)

[Kareem A. Eltouny](#), [Xiao Liang](#)

### [Multi-channel response reconstruction using transformer based generative adversarial network](#)

[Wenhao Zheng](#), [Jun Li](#), [Qilin Li](#), [Hong Hao](#)

### [Geometry-guided semantic segmentation for post-earthquake buildings using optical remote sensing images](#)

[Yu Wang](#), [Xin Jing](#), [Yang Xu](#), [Liangyi Cui](#), [Qiangqiang Zhang](#), [Hui Li](#)

### [Deep neural network-based regional seismic loss assessment considering correlation between EDP residuals of building structures](#)

[Chulyoung Kang](#), [Taeyong Kim](#), [Oh-Sung Kwon](#), [Junho Song](#)

### [Efficient regional seismic risk assessment via deep generative learning of surrogate models](#)

[Shanwu Li](#), [Charles Farrar](#), [Yongchao Yang](#)

### [Simulation-based methodology to identify damage indicators and safety thresholds for post-earthquake evaluation of structures](#)

[Francisco A. Galvis](#), [Anne M. Hulseley](#), [Jack W. Baker](#), [Gregory G. Deierlein](#)

### [Generalized stacked LSTM for the seismic damage evaluation of ductile reinforced concrete buildings](#)

[Bilal Ahmed](#), [Sujith Mangalathu](#), [Jong-Su Jeon](#)

### [Seismic damage prediction of RC buildings using machine learning](#)

[Sanjeev Bhatta](#), [Ji Dang](#)

# Review paper on EQ engineering

---



*remote sensing*

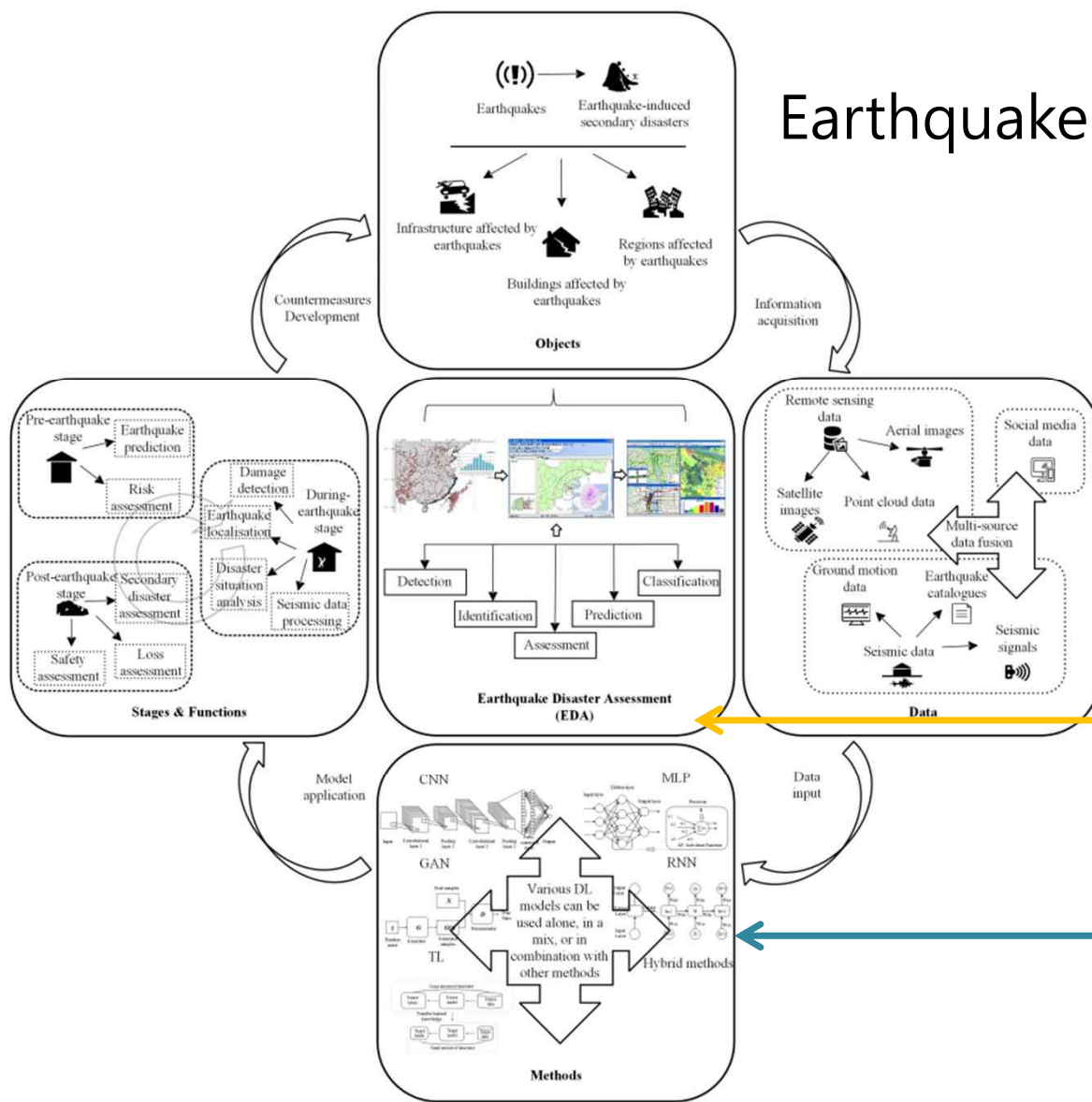
## Review paper

Deep Learning for Earthquake Disaster Assessment:  
Objects, Data, Models, Stages, Challenges, and  
Opportunities

Jing Jia and Wenjie Ye\*



# Earthquake Disaster Assessment Framework



Collection of Data

Objectives of ML and DL

Typical DL methods

Figure 6. The application framework of DL for EDA from four dimensions.

# Structural Health Monitoring

Large-scale structural health monitoring using composite recurrent neural networks and grid environments

By Kareem A. Eltouny Xiao Liang

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING



The framework relies on a 5D, time dependent grid environment and a novel spatiotemporal composite **autoencoder** network. This network is a hybrid of **autoencoder convolutional neural networks** and **long short-term memory networks**. A 10-story, 10-bay, numerical structure is used to evaluate the proposed framework damage diagnosis capabilities. The framework was successful in diagnosing the structure health state with average accuracies of 93% and 85% for damage detection and localization, respectively.

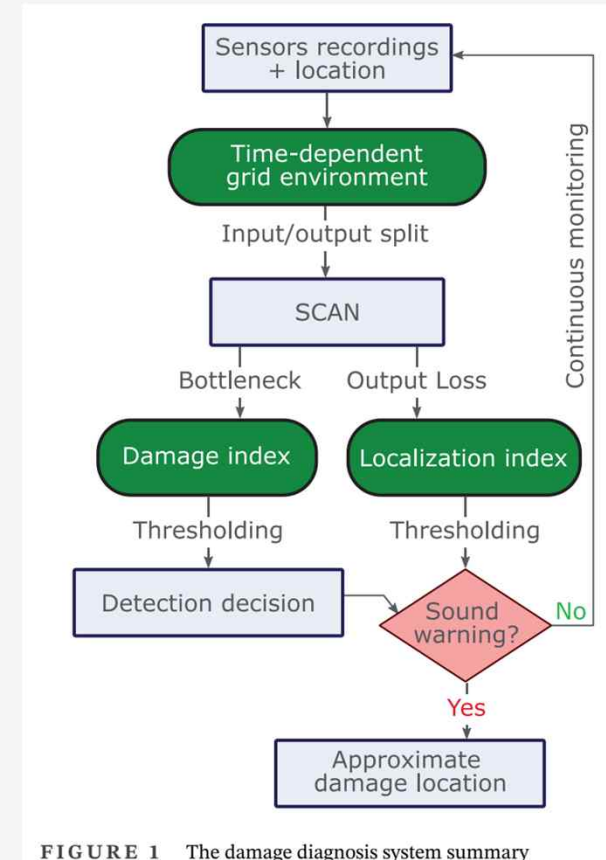


FIGURE 1 The damage diagnosis system summary

# Composite autoencoder architecture

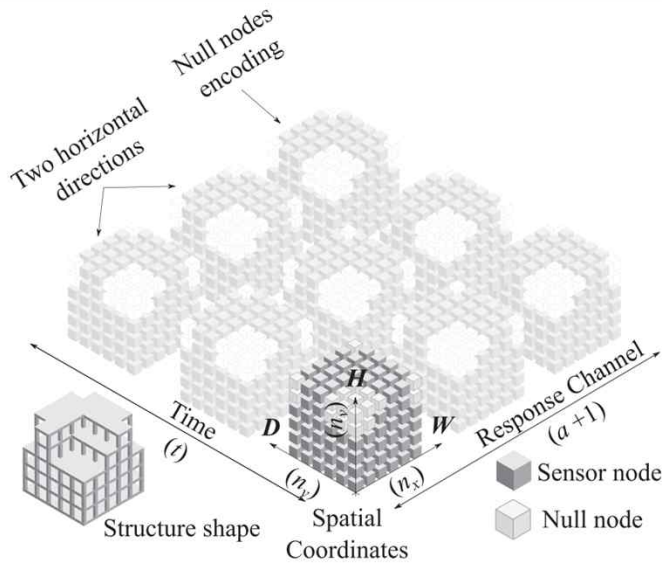
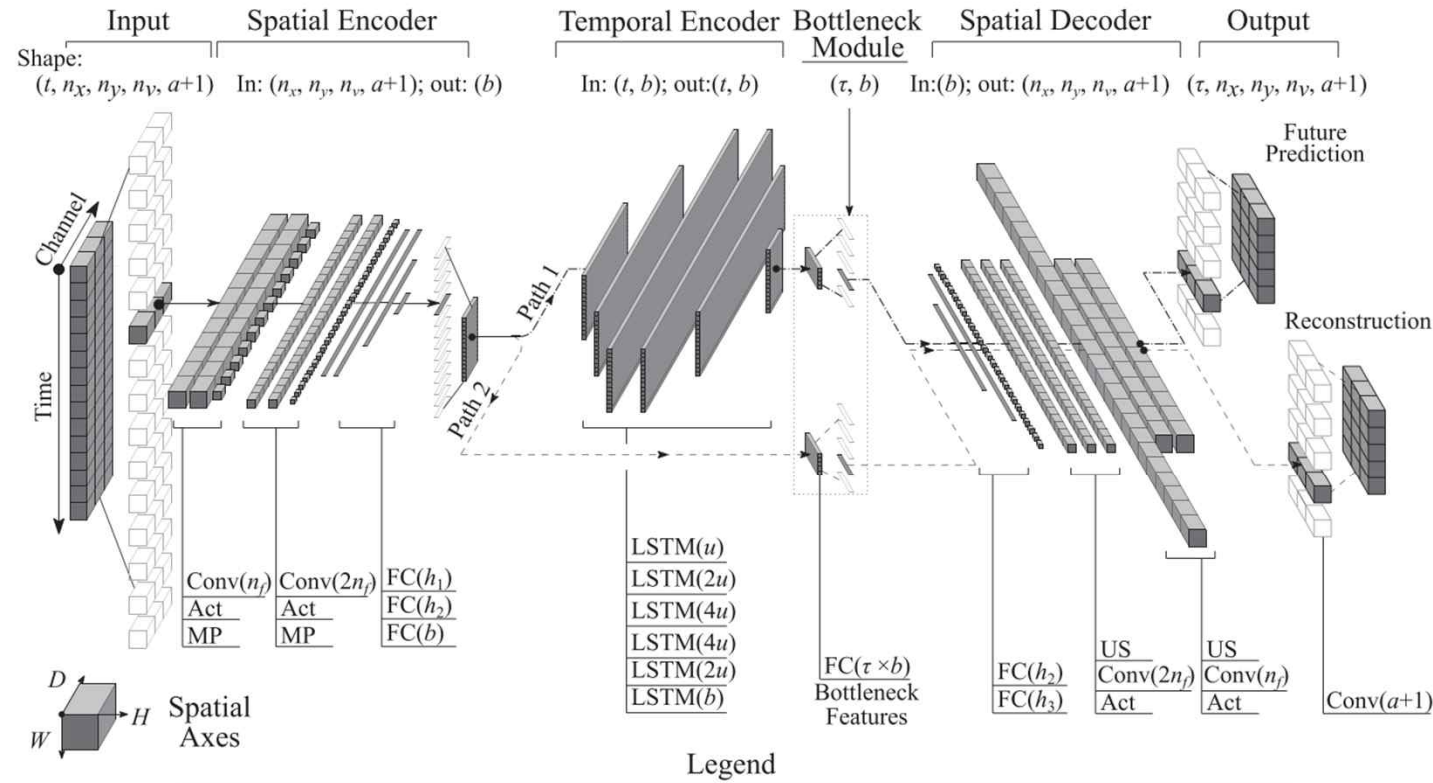


FIGURE 2 An example of the proposed time-dependent grid environment

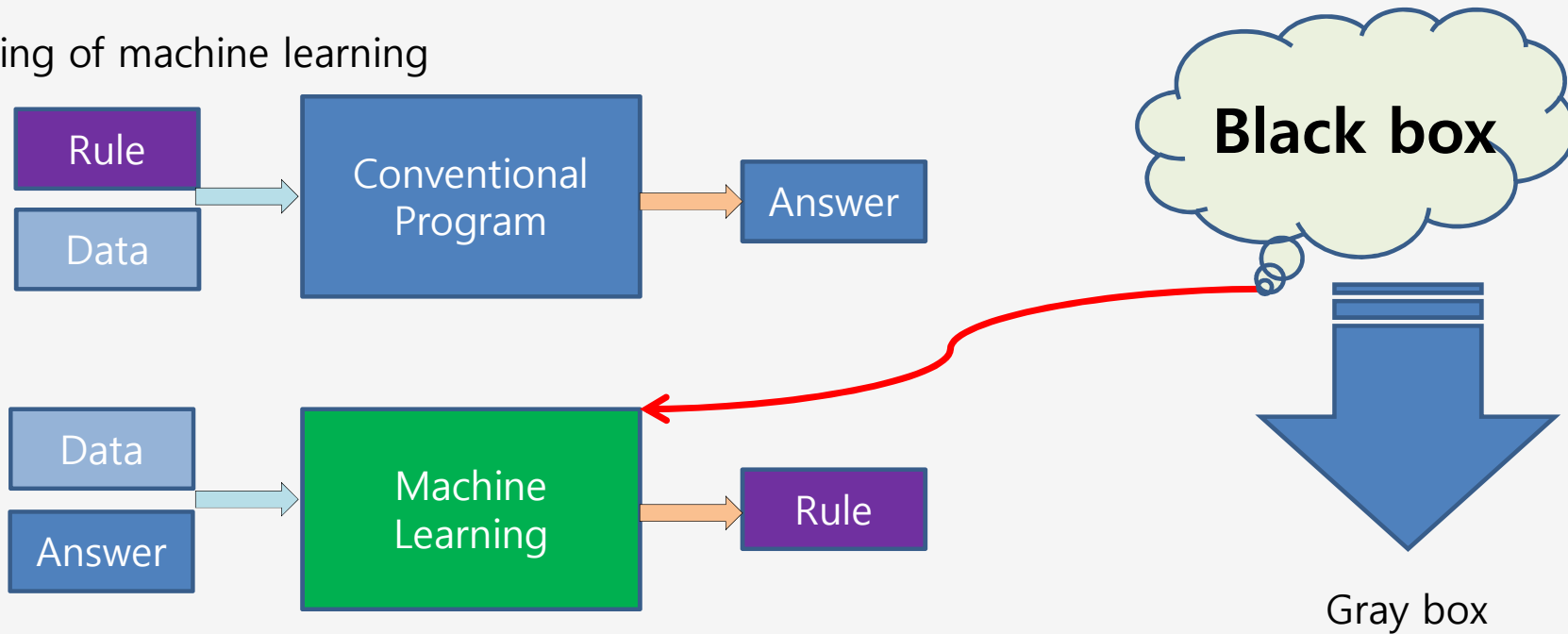


Conv( $n_f$ ): Convolution (filters)    FC( $h_1$ ): Fully connected (nodes)    LSTM( $u$ ): Long-short term memory (units)    Act: Activation  
 MP: Max pooling    US: Upsampling     $\bullet \dashrightarrow$  Future prediction path     $\bullet \dashrightarrow$  Reconstruction path     $\bullet \rightarrow$  Combined paths

FIGURE 3 The proposed spatiotemporal composite autoencoder network architecture

# Motivation

understanding of machine learning



**Difference between traditional approach and ML**

# Contents

- Regression
- Matrix for Optimization
- Diagonalization
- Image Compression and Compressed sensing
  
- Data-driven Machine learning
- Physic-informed neural network
- Neural operator
- Kernelization

# Basics in ML and DL

---

## Deterministic model

$$y = f(x; \theta)$$

$$x \in X, y \in Y, \text{ and } \theta = \{\theta_1, \dots, \theta_D\}$$

## Probabilistic model

Generative model distribution

$$p(x, y; \theta)$$

Discriminative model distribution

$$p(y|x; \theta)$$

# Summary of model functions and distribution

	Method	Model function/Distribution
Data fitting	Linear regression Nonlinear regression	$f(x; \mathbf{w}) = \mathbf{w}^T x$ $f(x; \mathbf{w}) = \varphi(\mathbf{w}, x)$
Artificial neural networks	Perception Feed-forward neural network Recurrent neural network Boltzmann machine	$f(x; \mathbf{w}) = \varphi(\mathbf{w}^T, x)$ $f(x; \mathbf{W}_1, \mathbf{W}_2, \dots) = \dots \varphi_2(\mathbf{W}_2 \varphi_1(\mathbf{W}_1 x)) \dots$ $f(x^{(t)}; \mathbf{W}) = \varphi(\mathbf{W} f(x^{(t-1)}; \mathbf{W}))$ $p(\mathbf{v}; \boldsymbol{\theta}) = \frac{1}{Z} \sum_h e^{-E(\mathbf{v}, h; \boldsymbol{\theta})}$
Graphical models	Bayesian network Hidden Markov model	$p(s) = \prod_k p(s_k   \pi_k; \boldsymbol{\theta})$ $p(V, O) = \prod_{t=1}^T p(v^{(t)}   v^{(t-1)}) \prod_{t=1}^T p(o^{(t)}   v^{(t)})$
Kernel methods	Kernel density estimation K-nearest neighbor Support vector machine Gaussian Process	$p(x y=c) = \frac{1}{M_c} \sum_{m y^m=c}^M k(x-x^m)$ $p(x y=c) = \frac{\#NN_c}{k}$ $f(x) = \sum_{m=1}^M \gamma_m y^m \kappa(x, x^m) + w_0$ $p(y x) = N[\tilde{\mathbf{k}}^T K^{-1} y; \tilde{\mathbf{k}} - \tilde{\mathbf{k}}^T K^{-1} \tilde{\mathbf{k}}]$

# Regression

---

$$Y = f(\mathbf{X}, \beta)$$

The general relationship between independent variables  $\mathbf{X}$ , dependent variables  $\mathbf{Y}$ , and some unknown parameter  $\beta$

Curve fitting results in **an optimization problem**.

The optimization can be mathematically framed as solving the linear system of equations

$$\mathbf{A} \mathbf{x} = \mathbf{b}$$

A simple solution for this linear problem uses the Moore-Penrose pseudo-inverse  $\mathbf{A}^\dagger$ .

$$\mathbf{x} = \mathbf{A}^\dagger \mathbf{b}$$



# Linear Prediction

After training and validation

$$y_i = \sum_{j=1}^d x_{ij} \beta_j$$

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta}$$

$$\mathbf{b} = \mathbf{A}\mathbf{x}$$

$$\mathbf{y} \in \mathbb{R}^{n \times 1}$$

prediction

$$\mathbf{X} \in \mathbb{R}^{n \times d}$$

Given data

$$\boldsymbol{\beta} \in \mathbb{R}^{d \times 1}$$

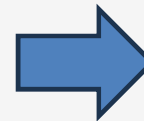
parameter

In matrix form

Given Data

$$\begin{bmatrix} t_1 \\ \cdot \\ \cdot \\ t_n \end{bmatrix} = \begin{bmatrix} x_{11} & \cdot & \cdot & x_{1d} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ x_{n1} & \cdot & \cdot & x_{nd} \end{bmatrix} \begin{bmatrix} \beta_1 \\ \cdot \\ \cdot \\ \beta_d \end{bmatrix}$$

Optimization



$$y_{\text{predict}} = \begin{Bmatrix} x_{\text{new},1} \\ \cdot \\ \cdot \\ x_{\text{new},d} \end{Bmatrix}^T \begin{bmatrix} \beta_1 \\ \cdot \\ \cdot \\ \beta_d \end{bmatrix}$$

label

Feature vector

Learnable parameter

# MSE cost function for Linear regression model

---

$$\text{MSE}(\mathbf{X}, h_{\theta}) = \frac{1}{m} \sum_{i=1}^m \left( \boldsymbol{\theta}^T \mathbf{x}^{(i)} - t^{(i)} \right)^2$$

Normal Equation to find the value of  $\theta$  that minimizes the cost function

$$\hat{\boldsymbol{\theta}} = \left( \mathbf{X}^T \mathbf{X} \right)^{-1} \mathbf{X}^T \mathbf{t}$$

```
X_b = np.c_[np.ones((100, 1)), X] # add x0 = 1 to each instance  
theta_best = np.linalg.inv(X_b.T.dot(X_b)).dot(X_b.T).dot(y)
```

↓

$$\hat{\boldsymbol{\theta}} = \mathbf{X}^+ \mathbf{t}$$

$$\mathbf{X}^+ = \mathbf{V} \boldsymbol{\Sigma}^+ \mathbf{U}^T \quad \text{Singular Value Decomposition}$$

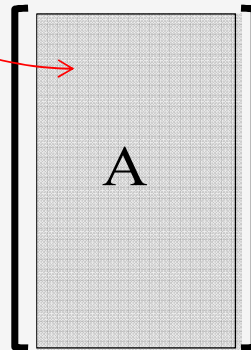
# Matrix for Optimization

given data or operator

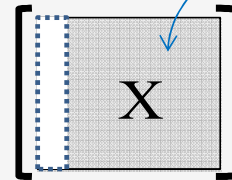
## Overdetermined Problem

Zero solution for **A**

Model terms

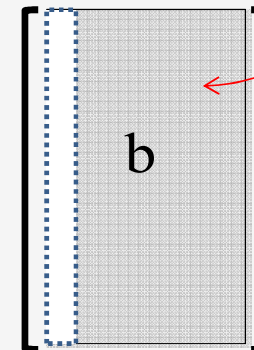


Loadings



=

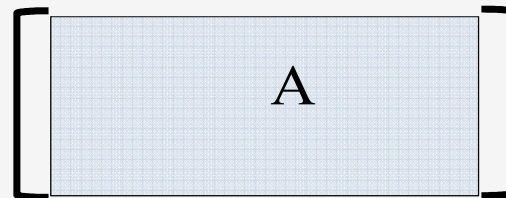
Outcomes



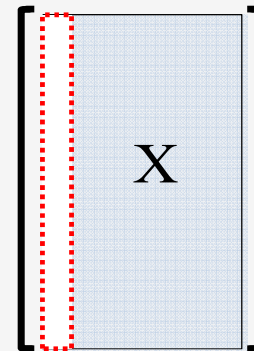
Target given

## Underdetermined Problem

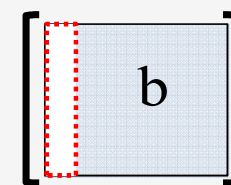
Too many solution for **A**



Feature: dimension



=



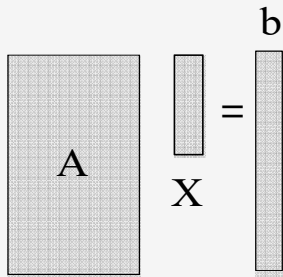
# Solution $\mathbf{X}$

$$\mathbf{Ax}=\mathbf{b} \quad \text{Non-square } A$$

How to deal with non-square matrices  $\rightarrow$  SVD, PCA,.....

Overdetermined,  $n>m$  (tall skinny  $A$ )

Many data


$$\mathbf{A} \mathbf{x} = \mathbf{b}$$

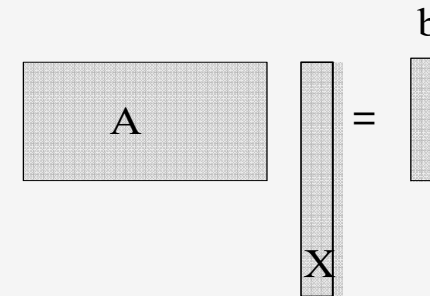
Zero solution  $\mathbf{x}$  for given  $\mathbf{b}$

$$\min \|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\|_2$$

**Optimization**

$$\operatorname{argmin} (\|\mathbf{Ax} - \mathbf{b}\|_2 + \lambda g(\mathbf{x}))$$

Underdetermined,  $n<m$  (short fat  $A$ )


$$\mathbf{A} \mathbf{x} = \mathbf{b}$$

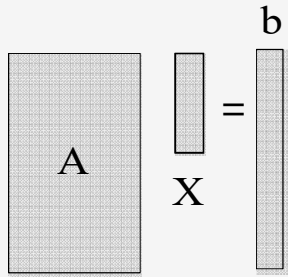
Too many solution  $\mathbf{x}$  for given  $\mathbf{b}$

$$\min \|\tilde{\mathbf{x}}\|_2 \quad \text{such that } \mathbf{A}\tilde{\mathbf{x}}=\mathbf{b}$$

**Optimization**

$$\operatorname{argmin} g(\mathbf{x}) \quad \text{subject to } \|\mathbf{Ax} - \mathbf{b}\|_2 \leq \varepsilon$$

# Over-determined Systems



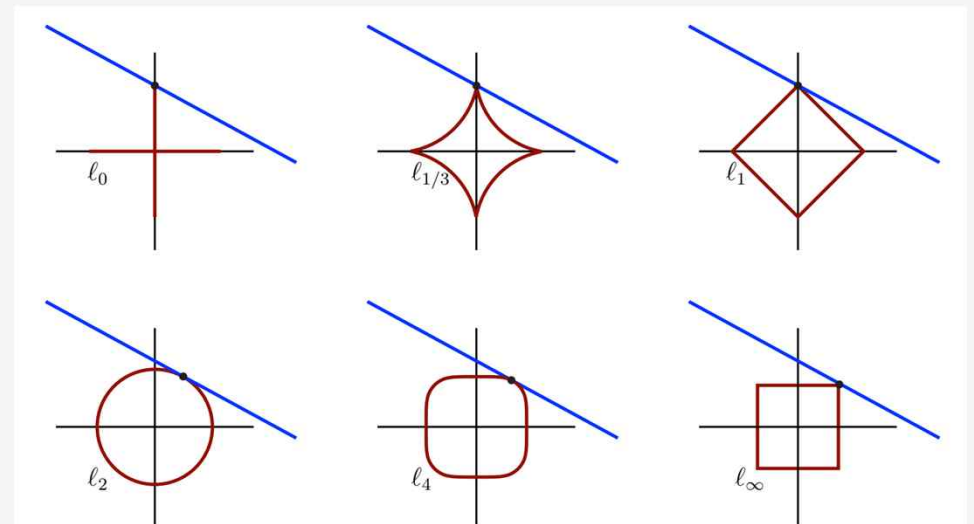
Zero solution  $\mathbf{x}$  for given  $\mathbf{b}$

$$\min \|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\|_2$$

Optimization

$$\hat{\mathbf{x}} = \underset{\substack{l_2 \text{ norm} \\ \text{Lasso} \quad \text{Ridge}}}{\operatorname{argmin}} \left\| \mathbf{A}\mathbf{x} - \mathbf{b} \right\|_2 + \lambda_1 \|\mathbf{x}\|_1 + \lambda_2 \|\mathbf{x}\|_2$$

Least absolute shrinkage and selection operator



The minimum-norm point on a line in different  $l_p$  norm

From Fig 3.10, p 110 in Ref.1

# Under-determined Systems

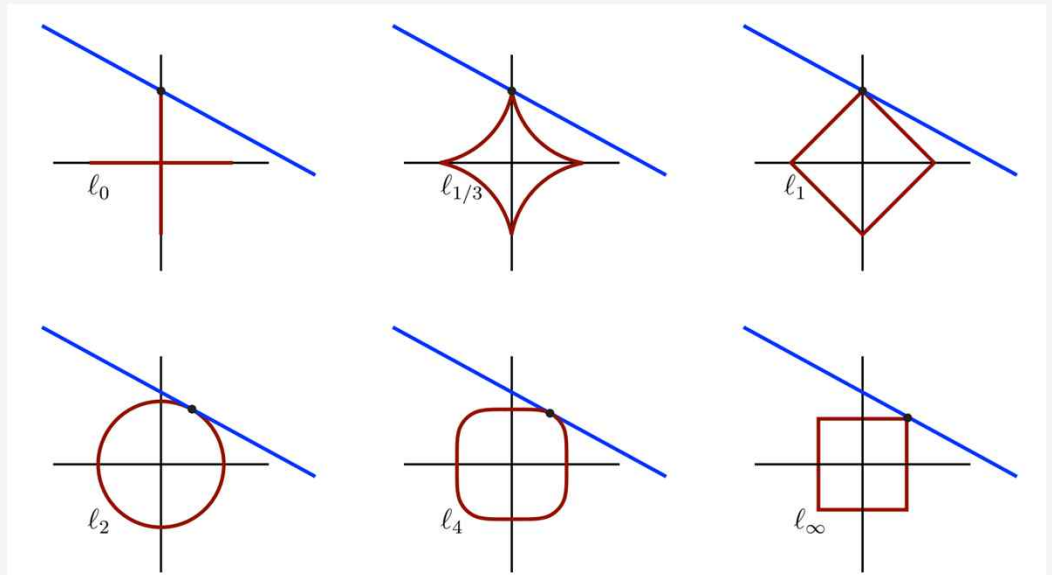
$$\begin{matrix} \boxed{\text{A}} \\ \text{X} \end{matrix} = \begin{matrix} \text{b} \\ \end{matrix}$$

Too many solution  $\mathbf{x}$  for given  $\mathbf{b}$

$$\min \|\tilde{\mathbf{x}}\|_p \quad \text{subject to } \mathbf{A}\tilde{\mathbf{x}} = \mathbf{b}$$

## Optimization

$$\min \left( \lambda_1 \|\mathbf{x}\|_1 + \lambda_2 \|\mathbf{x}\|_2 \right) \quad \text{subject to } \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 \leq \varepsilon$$



# Diagonalization (Square matrix)

$$\underline{\underline{\mathbf{A}}}\underline{\underline{\mathbf{v}}} = \lambda \underline{\underline{\mathbf{v}}}$$

Eigen vector

Eigen value

$$\mathbf{A} = \mathbf{X}\mathbf{\Lambda}\mathbf{X}^{-1}$$

$$\mathbf{X}^{-1}\mathbf{A}\mathbf{X} = \mathbf{\Lambda}$$

$$[\mathbf{A}] = \begin{bmatrix} 5 & 1 \\ 3 & 3 \end{bmatrix} \Rightarrow |\mathbf{A} - \lambda\mathbf{I}| = 0$$

$$\vec{v}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \lambda_1 = 6 \quad \vec{v}_2 = \begin{bmatrix} 1 \\ -3 \end{bmatrix}, \lambda_2 = 2$$

$$\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \Rightarrow \mathbf{A}^{-1} = \frac{1}{ad - cb} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} 1 & 1 \\ 1 & -3 \end{bmatrix} \quad \mathbf{X}^{-1} = \frac{1}{4} \begin{bmatrix} 3 & 1 \\ 1 & -1 \end{bmatrix}$$

$$\mathbf{X}^{-1}\mathbf{A}\mathbf{X} = \frac{1}{4} \begin{bmatrix} 3 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 5 & 1 \\ 3 & 3 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -3 \end{bmatrix} = \begin{bmatrix} 6 & 0 \\ 0 & 2 \end{bmatrix}$$

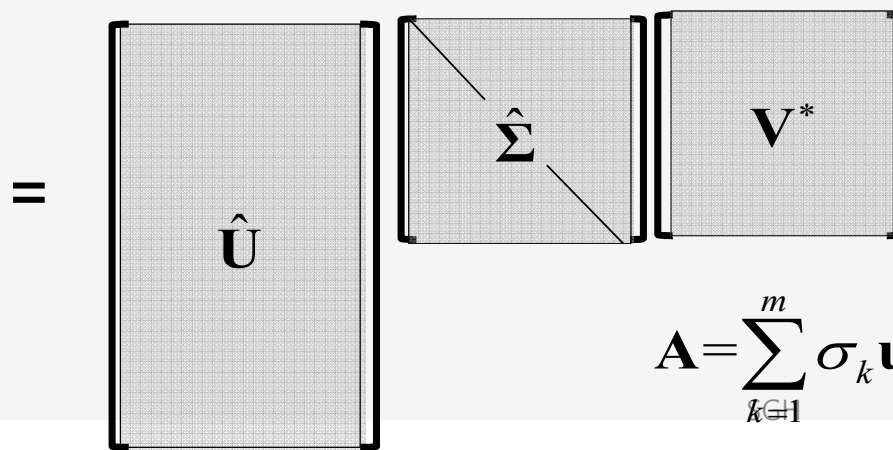
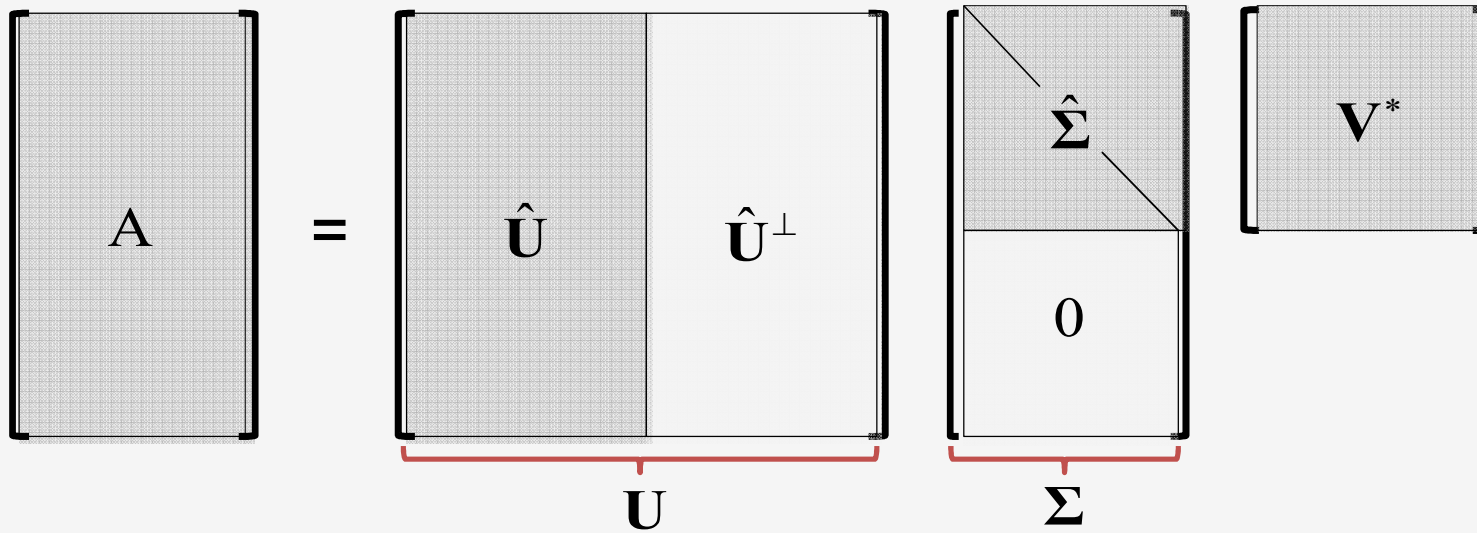
$$[\mathbf{A}] = \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix} \Rightarrow |\mathbf{A} - \lambda\mathbf{I}| = 0$$

$$\vec{v}_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \lambda_1 = 4 \quad \vec{v}_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \lambda_2 = 2$$

$$\mathbf{X} = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \quad \mathbf{X}^{-1} = \frac{1}{2} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} = \mathbf{X}^T$$

$$\mathbf{X}^{-1}\mathbf{A}\mathbf{X} = \frac{1}{2} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 0 \\ 0 & 2 \end{bmatrix}$$

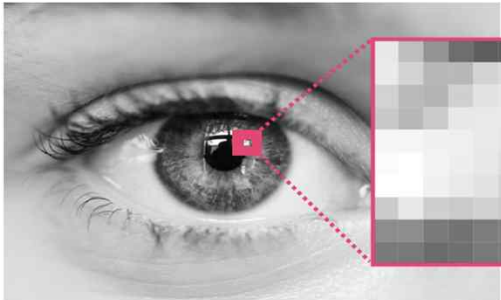
# Rectangular matrix decomposition(SVD)



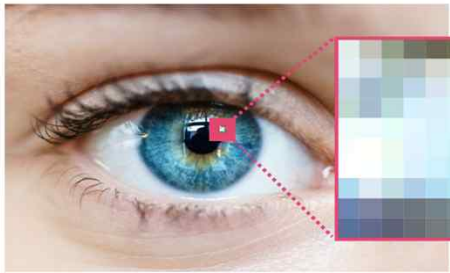
$$A = \sum_{k \in I}^m \sigma_k \mathbf{u}_k \mathbf{v}_k^* = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^* + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^* + \cdots + \sigma_m \mathbf{u}_m \mathbf{v}_m^*$$



# Image Compression and Compressed Sensing



230	194	147	108	90	98	84	96	91	101
237	206	188	195	207	213	163	123	116	128
210	183	180	205	224	234	188	122	134	147
198	189	201	227	229	232	200	125	127	135
249	241	237	244	232	226	202	116	125	126
251	254	241	239	230	217	196	102	103	99
243	255	240	231	227	214	203	116	95	91
204	231	208	200	207	201	200	121	95	95
144	140	120	115	125	127	143	118	92	91
121	121	108	109	122	121	134	106	86	97



											233	188	137	96	90	95	63	73	73	82	
											237	202	159	120	105	110	88	107	112	121	109
226	191	147	110	101	112	98	123	110	119	142	131										
221	191	176	182	203	214	169	144	133	145	155	122										
185	160	161	184	205	223	186	137	147	161	140	115										
181	174	189	207	206	215	194	136	142	151	133	87										
246	237	237	231	208	206	192	122	143	144	111	74										
254	254	241	224	199	192	181	99	122	117	107	74										
239	248	232	207	187	182	184	110	114	110	113	74										
193	215	193	167	158	164	181	114	112	111	105	82										
113	119	110	111	113	123	135	120	108	106	113											
93	97	91	103	107	111	122	112	104	114												

# Image Compression

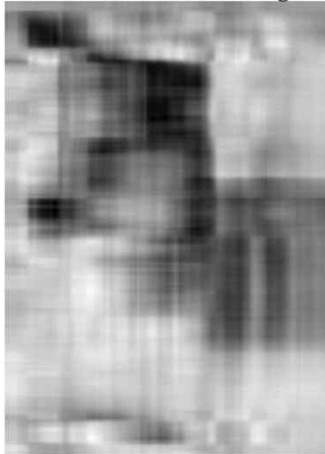
Original



$r = 20$ , 2.33% storage



$r = 5$ , 0.57% storage



$r = 100$ , 11.67% storage



$$\mathbf{X} = \sum_{k=1}^m \sigma_k \mathbf{u}_k \mathbf{v}_k^* = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^* + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^* + \cdots + \sigma_m \mathbf{u}_m \mathbf{v}_m^*$$



$$\mathbf{X} - \tilde{\mathbf{X}} = \sum_{k=r+1}^m \sigma_k \mathbf{u}_k \mathbf{v}_k^*$$

Image compression truncating the SVD at various rank  $r$   
Original image resolution is 2000 X 1500

From Fig 1.3, p 110 in Ref.1

# 2D Fourier Transformation for Image

$$\hat{f}(\omega) = F(f(x)) = \int_{-\infty}^{\infty} f(x) e^{-i\omega x} dx$$

$$f(x) = F^{-1}(\hat{f}(\omega)) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(\omega) e^{i\omega x} d\omega$$



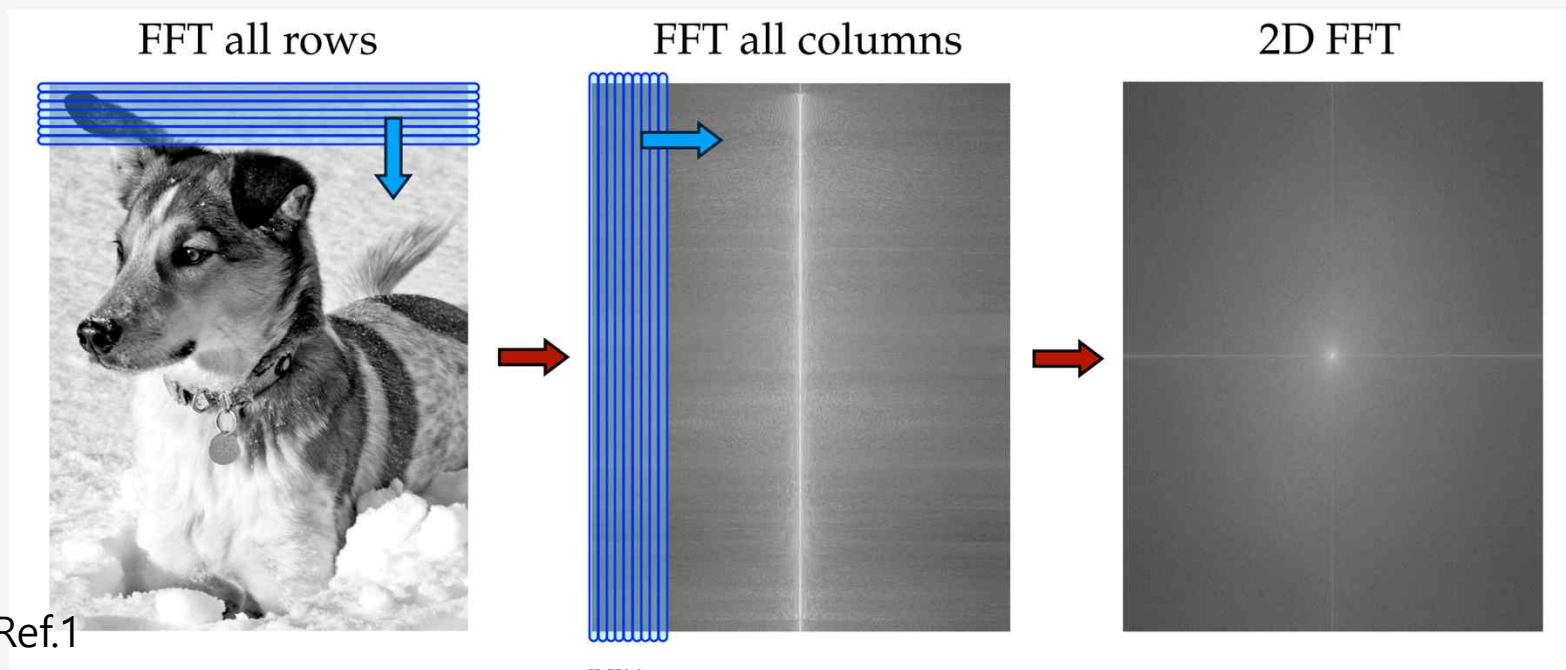
$$\hat{f}(u, v) = F(f(x, y)) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-i\omega(ux+vy)} dx dy$$

$$f(x, y) = F^{-1}(\hat{f}(u, v)) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \hat{f}(u, v) e^{i\omega(ux+vy)} du dv$$

Image compression  
truncating Fourier  
Transformation

Original image  
resolution is 2000 X  
1500

From Fig 1.3, p 110 in Ref.1



Full image



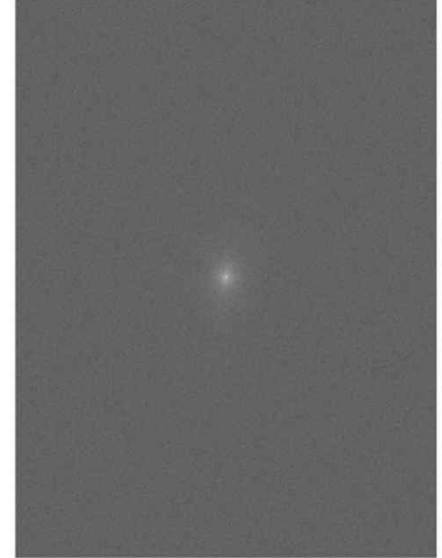
5.0% of FFT



Noisy image



Noisy FFT



1.0% of FFT



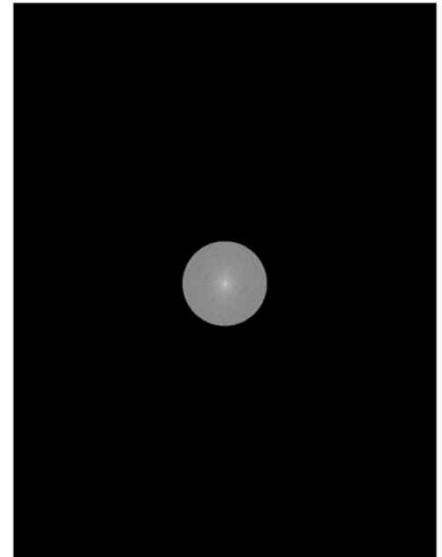
0.2% of FFT



Filtered image



Filtered FFT



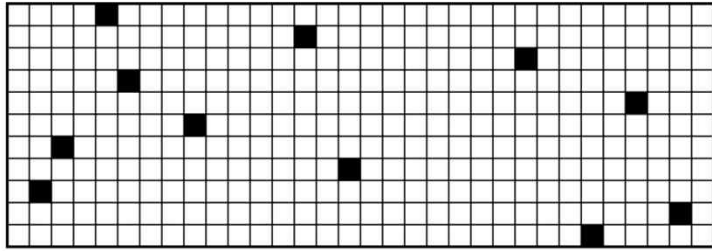
# Compressed sensing



$y$



=

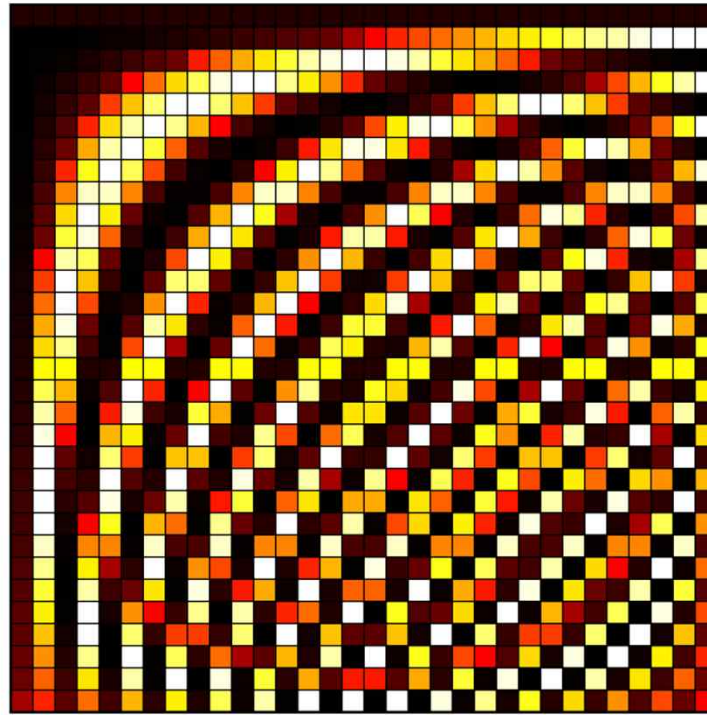


measurement

Measurement matrix

$C$

$\Psi$



Orthonormal basis

$s$



Sparse vector

Schematic of measurement in compressed sensing framework

From Fig 3.4, p 103 in Ref.1

$$y = Cx$$

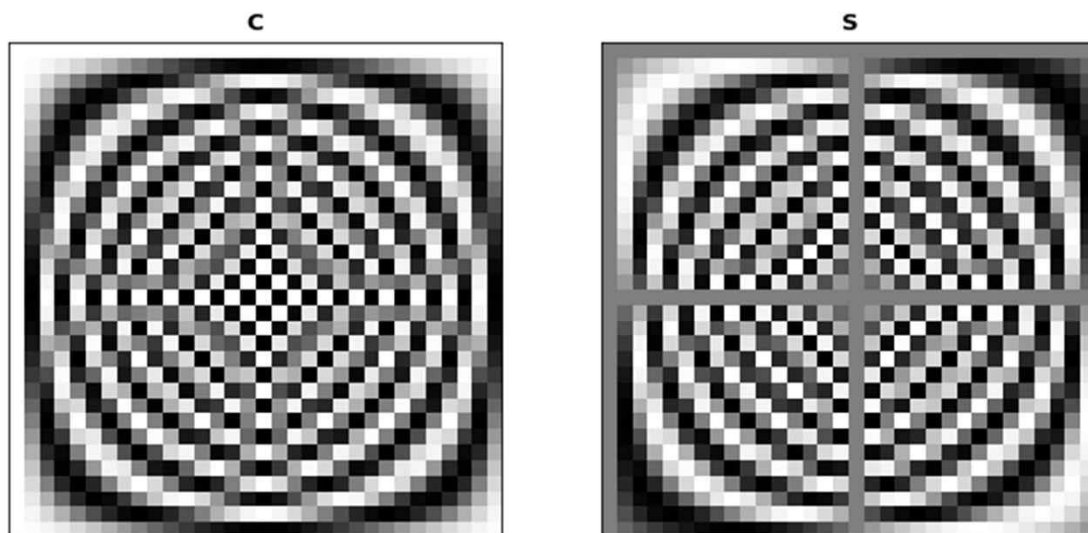
$$y = \underbrace{C\Psi}_{\Theta} s = \Theta s$$

$$\hat{s} = \arg \min \|s\|_0$$

subject to  $y = C\Psi s$

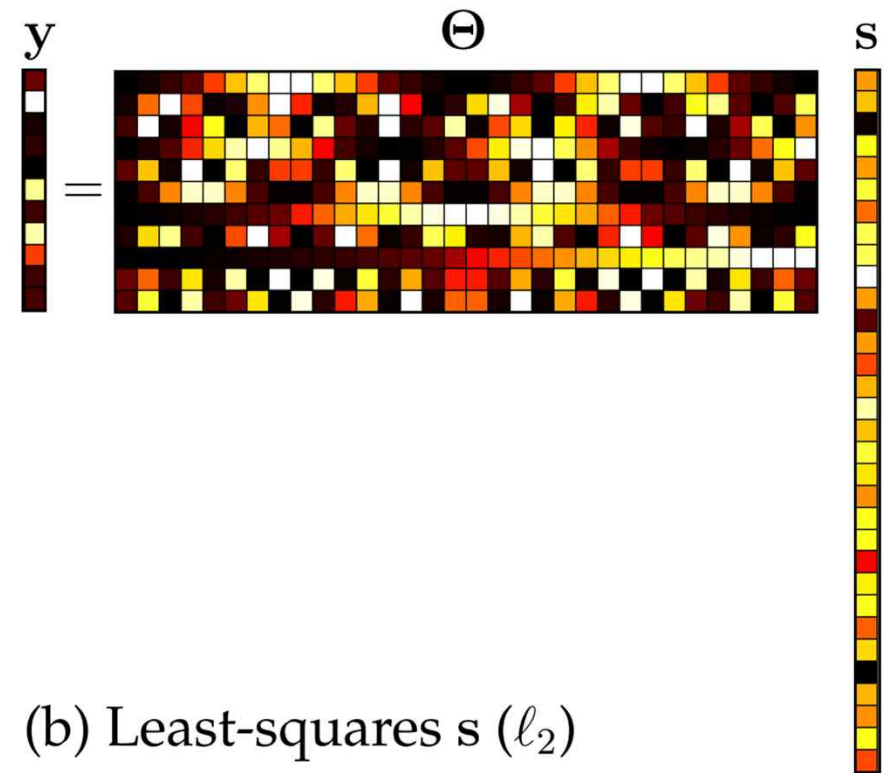
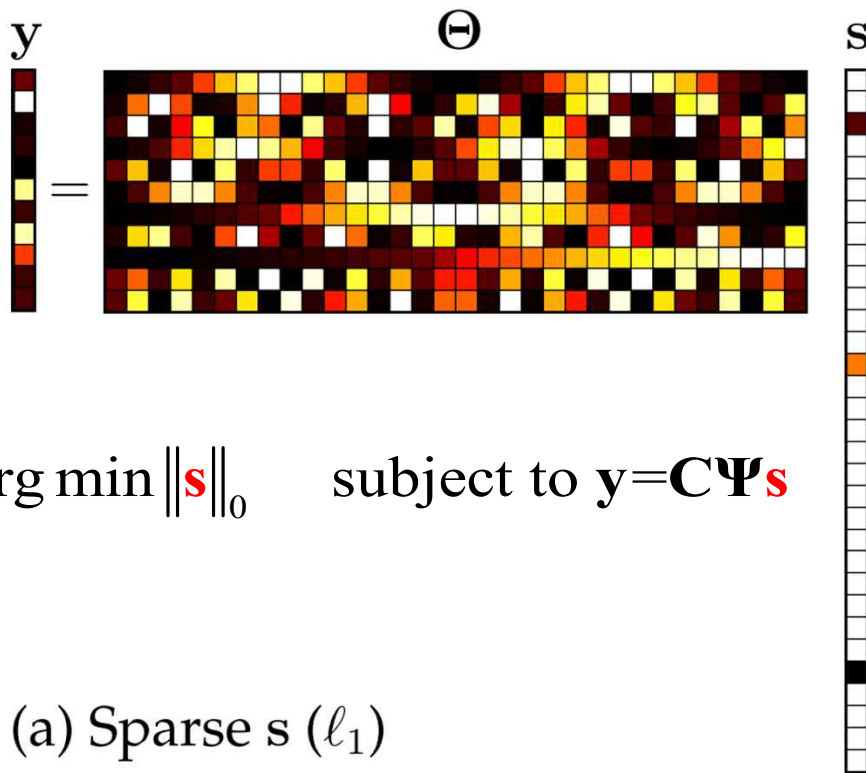
# Fourier Basis; Orthonormal basis

---



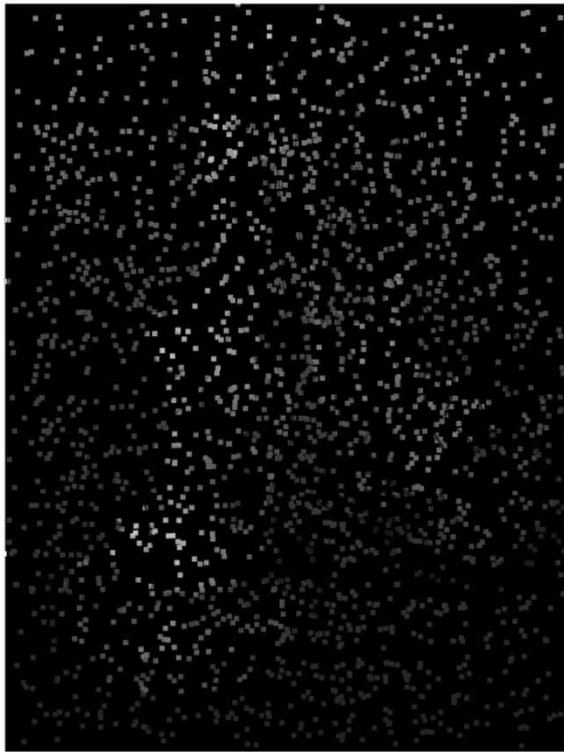
[The Fourier basis — Tutorials on imaging, computing and mathematics \(matthew-brett.github.io\)](https://matthew-brett.github.io)

# Solutions dependent on norm selection

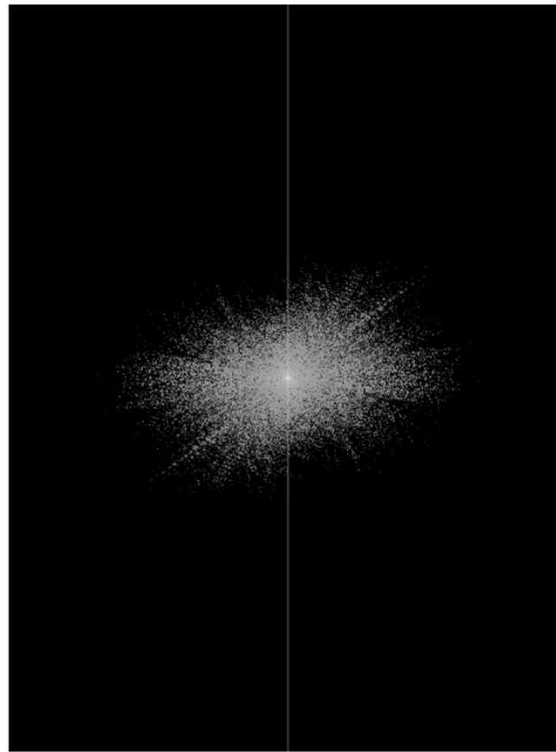


From Fig 3.5, p 103 in Ref.1

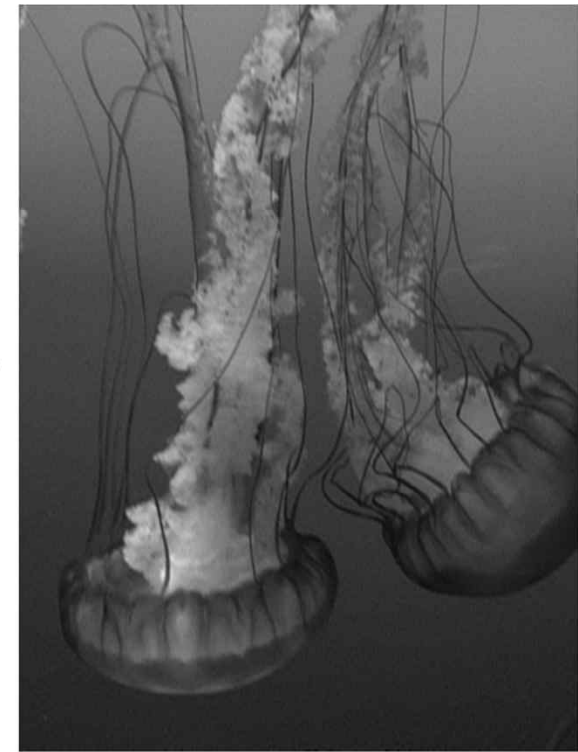
Measurements,  $y$



Sparse coefficients,  $s$



Reconstructed image,  $x$



$\xrightarrow{\ell_1}$

$\xrightarrow{\mathcal{F}^{-1}}$

Measurement

Sparse coefficient

Reconstructed image

$$p \approx O(K \ln(n/K)) \approx k_1 K \ln(n/K)$$

$$p \approx 3 * 0.05 * \underbrace{1024 * 768}_{\text{Pixel : } n} * \ln(20) = 353,390$$

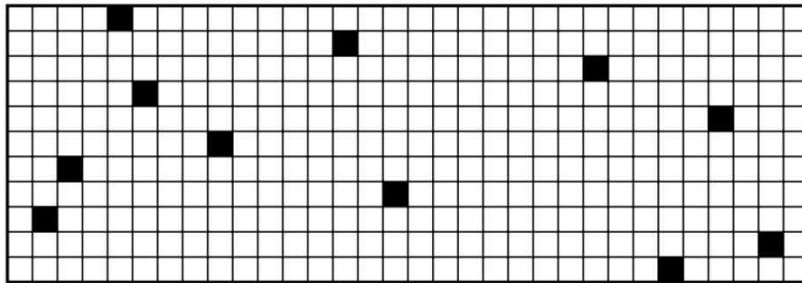
$$K = 0.05 * 1024 * 768$$

SGH

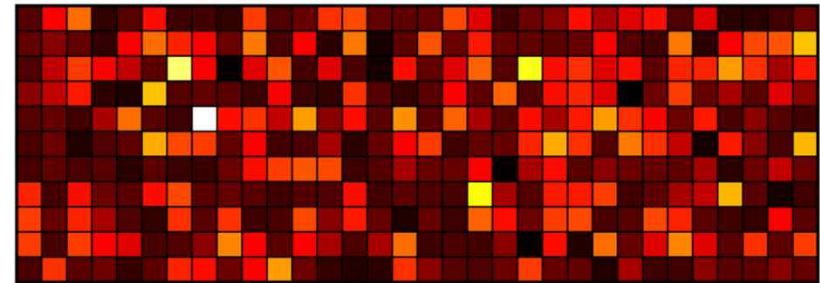


# Measurement matrices

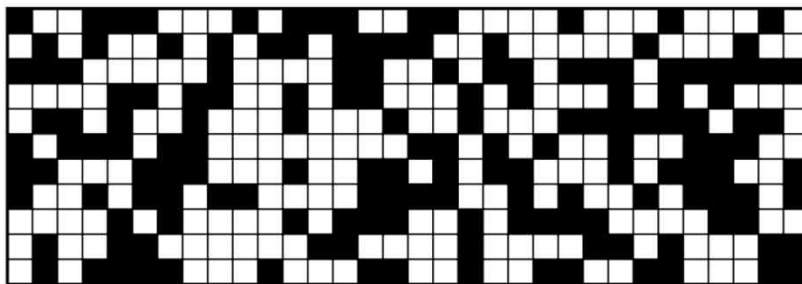
(a) Random single pixel



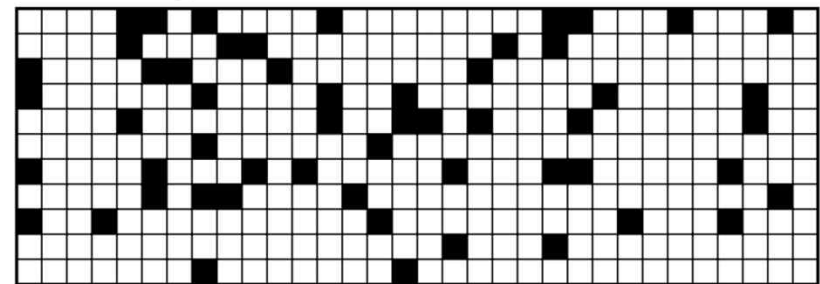
(b) Gaussian random



(c) Bernoulli random



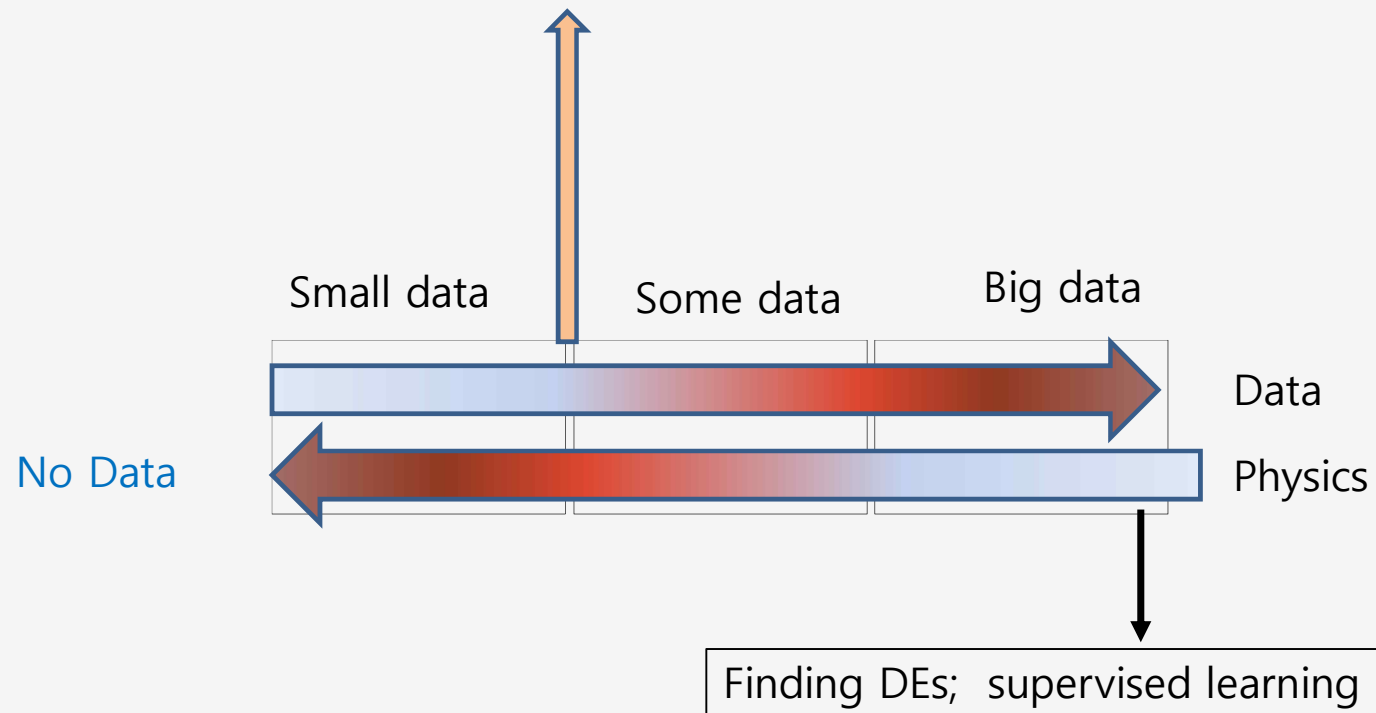
(d) Sparse random



**Examples of good random measurement matrices**

From Fig 3.11, p 112 in Ref.1

# Data-Driven Machine Learning



# Agenda for Engineering with AI

---

## PHYSICS INFORMED MACHINE LEARNING

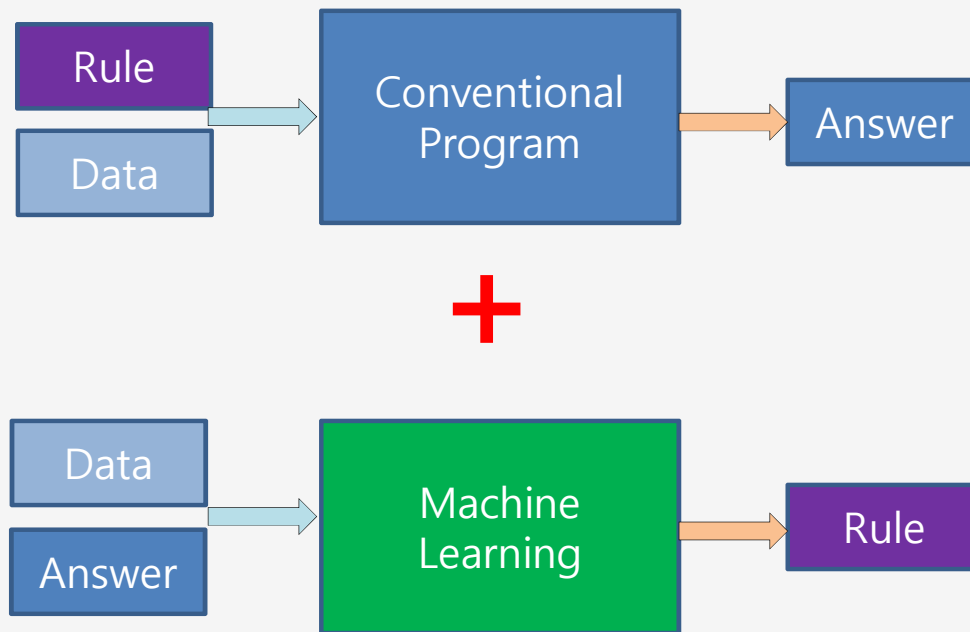
- Derivation of Governing Equations  
**Data-driven M/L**
- Solution of Governing Equations (PDE, DE)  
**Physics-informed M/L**

## PHYSICAL MODELS FROM DATA via OPTIMIZATION

- How to optimize by a few data
- Data-driven M/L -> to find linear operator

# Paradigm shift

---

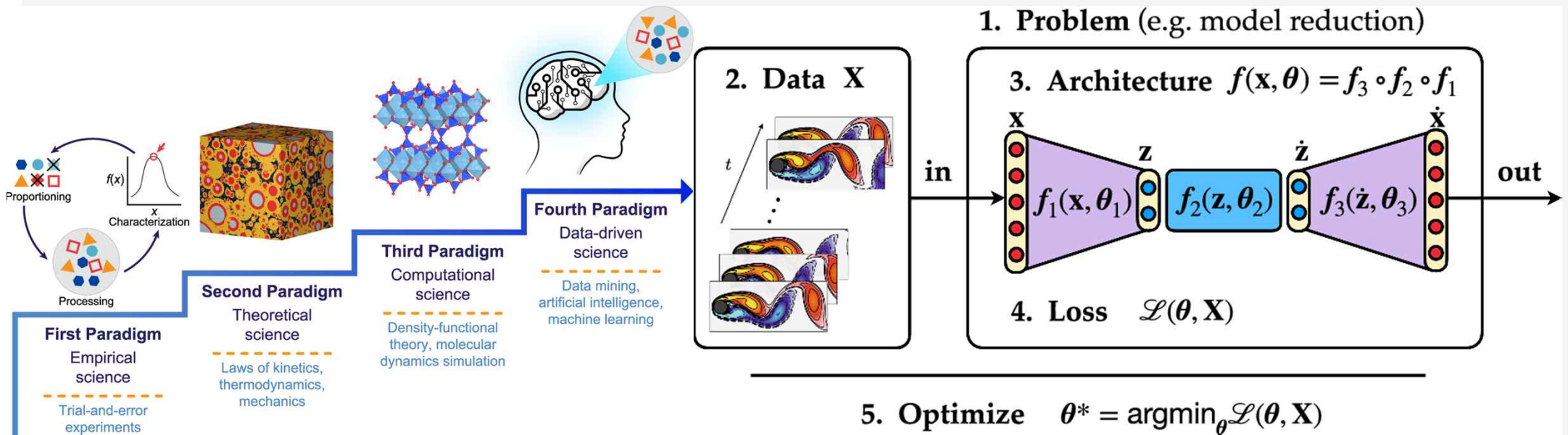


**Combination between traditional approach and ML**

# Application area of Data-driven ML

ML for new material

Fluid mechanics



[https://media.springernature.com/full/springer-static/image/art%3A10.1038%2Fs41524-022-00810-x/MediaObjects/41524\\_2022\\_810\\_Fig1\\_HTML.png?as=webp](https://media.springernature.com/full/springer-static/image/art%3A10.1038%2Fs41524-022-00810-x/MediaObjects/41524_2022_810_Fig1_HTML.png?as=webp)

Applying machine learning to study fluid mechanics

Invited Review

[Open access](#)

Published: 04 January 2022

Volume 37, pages 1718–1726, (2021)

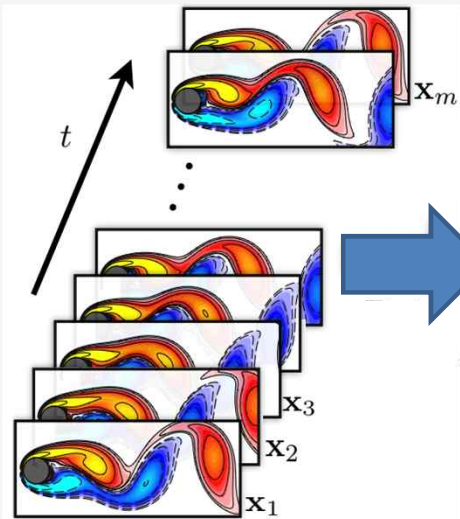
# Dynamic system

$$\frac{d}{dt} \mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t), t; \beta)$$

Vector field

State of system

Set of parameters



Linear dynamics and Spectral Decomposition

$$\frac{d}{dt} \mathbf{x} = \mathbf{A}\mathbf{x} \quad \longrightarrow \quad \mathbf{x}(t_0 + t) = e^{\mathbf{A}t} \mathbf{x}(t_0)$$

The dynamics are entirely characterized by the eigenvalues of  $\mathbf{A}$ , given by the spectral decomposition

$$\mathbf{A}\mathbf{T} = \mathbf{T}\mathbf{\Lambda}$$

$$\mathbf{A} = \mathbf{T}\mathbf{\Lambda}\mathbf{T}^{-1}$$

$$\mathbf{x}(t_0 + t) = \mathbf{T}e^{\mathbf{\Lambda}t}\mathbf{T}^{-1}\mathbf{x}(t_0)$$

Transformation of coordinate gives decoupled system

$$\mathbf{z} = \mathbf{T}^{-1}\mathbf{x}$$

$$\frac{d}{dt} \mathbf{z} = \mathbf{\Lambda}\mathbf{z}$$

# Dynamic Mode Decomposition

---

$$\mathbf{X} = \begin{bmatrix} \parallel & \parallel & \cdots & \cdots & \parallel \\ \mathbf{x}(t_1) & \mathbf{x}(t_2) & & & \mathbf{x}(t_m) \\ \parallel & \parallel & & & \parallel \end{bmatrix}$$

$$\mathbf{X}' = \begin{bmatrix} \parallel & \parallel & \cdots & \cdots & \parallel \\ \mathbf{x}(t'_1) & \mathbf{x}(t'_2) & & & \mathbf{x}(t'_m) \\ \parallel & \parallel & & & \parallel \end{bmatrix}$$

$$\mathbf{X}' \approx \mathbf{A}\mathbf{X}$$

pseudo-inverse .

The best-fit operator  $\mathbf{A} = \arg \min_A \|\mathbf{X}' - \mathbf{A}\mathbf{X}\|_F = \mathbf{X}'\mathbf{X}^\dagger$

# Steps for DMD

Step 1. Compute the singular value decomposition of  $X$

$$\mathbf{X} \approx \tilde{\mathbf{U}} \tilde{\Sigma} \tilde{\mathbf{V}}^* \quad \text{reduced singular vector, conjugate transpose}$$

Step 2. The reduced-order matrix  $A$

$$\mathbf{A} = \mathbf{X}' \tilde{\mathbf{V}} \tilde{\Sigma}^{-1} \tilde{\mathbf{U}}^* \Rightarrow \tilde{\mathbf{A}} = \tilde{\mathbf{U}}^* \mathbf{A} \tilde{\mathbf{U}} = \tilde{\mathbf{U}}^* \mathbf{X}' \tilde{\mathbf{V}} \tilde{\Sigma}^{-1}$$

Step 3. The spectral decomposition of the **reduced matrix  $A$**

$$\tilde{\mathbf{A}} \mathbf{W} = \mathbf{W} \Lambda$$

$$\begin{aligned} \mathbf{A} \Phi &= \left( \overbrace{\mathbf{X}' \tilde{\mathbf{V}} \tilde{\Sigma}^{-1} \tilde{\mathbf{U}}^*}^{\mathbf{A}} \right) \left( \overbrace{\mathbf{X}' \tilde{\mathbf{V}} \tilde{\Sigma}^{-1} \mathbf{W}}^{\Phi} \right) \\ &= \mathbf{X}' \tilde{\mathbf{V}} \tilde{\Sigma}^{-1} \underbrace{\tilde{\mathbf{A}} \mathbf{W}}_{\tilde{\mathbf{A}}} \\ &= \mathbf{X}' \tilde{\mathbf{V}} \tilde{\Sigma}^{-1} \mathbf{W} \Lambda \\ &= \Phi \Lambda \end{aligned}$$

Step 4. The **high-dimensional DMD modes** are reconstructed

$$\Phi = \mathbf{X}' \tilde{\mathbf{V}} \tilde{\Sigma}^{-1} \mathbf{W}$$



---

## Algorithm 1 Exact DMD [4]

---

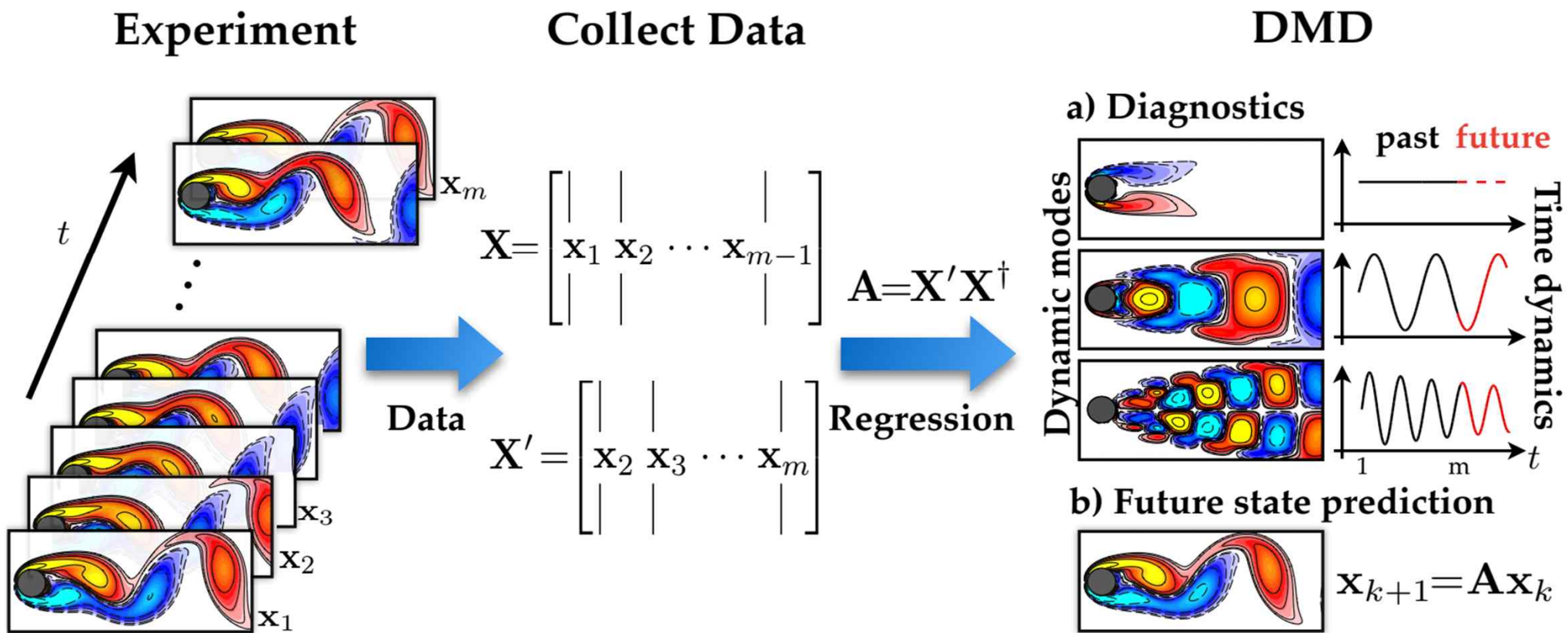
**Input:** Data matrix  $X$ , shifted data matrix  $X'$ , and target rank  $r$ .

**Output:** DMD spectrum  $\Lambda$  and modes  $\Phi$ .

- 1: **procedure** DMD ( $X, X', r$ )
- 2:      $[U, \Sigma, V] \leftarrow \text{SVD}(X, r)$                       $\triangleright$  Truncated  $r$ -rank SVD of  $X$ .
- 3:      $\tilde{A} \leftarrow U^* X' V \Sigma^{-1}$                       $\triangleright$  Low-rank approximation of  $A$ .
- 4:      $[W, \Lambda] \leftarrow \text{EIG}(\tilde{A})$                       $\triangleright$  Eigendecomposition of  $\tilde{A}$ .
- 5:      $\Phi \leftarrow X' V \Sigma^{-1} W$                       $\triangleright$  DMD modes of  $A$ .
- 6: **end procedure**

Note that if  $\lambda_i = 0$ , then  $\phi_i = U w_i$  for step 5. In the original DMD algorithm [57] all modes are computed as  $\phi_i = U w_i$ .

---



Overview of DMD illustrated on the fluid flow past a circular cylinder at Reynolds number 100. From Ref. 1

# Sparse Identification of Non-linear Dynamics

$$\frac{d}{dt} \mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t), t; \beta)$$

$$\mathbf{X} = [\mathbf{x}(t_1) \quad \mathbf{x}(t_2) \quad \cdots \quad \mathbf{x}(t_m)]^T$$

We seek to approximate  $\mathbf{f}$  by [a generalized linear model](#)

$$\mathbf{f}(\mathbf{x}) \approx \sum_{k=1}^p \theta_k(\mathbf{x}) \xi_k = \Theta(\mathbf{x}) \xi$$

$$\dot{\mathbf{X}} = [\dot{\mathbf{x}}(t_1) \quad \dot{\mathbf{x}}(t_2) \quad \cdots \quad \dot{\mathbf{x}}(t_m)]^T$$

A library of [candidate nonlinear function](#) may be constructed from the data in  $\mathbf{X}$

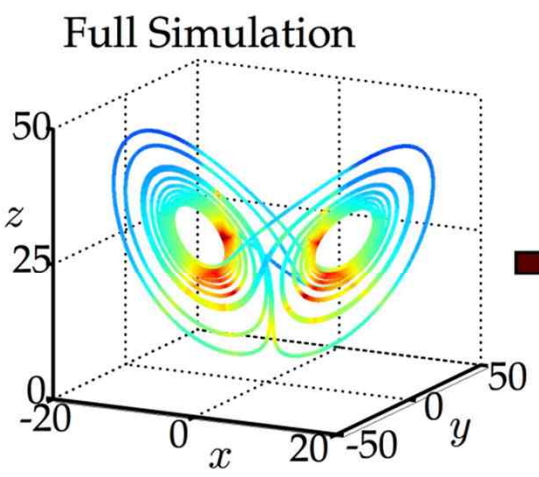
$$\Theta(\mathbf{X}) = [\mathbf{1} \quad \mathbf{X} \quad \mathbf{X}^2 \quad \cdots \quad \mathbf{X}^d \quad \cdots \quad \sin(\mathbf{X}) \quad \cdots]$$

$$\dot{\mathbf{X}} = \Theta(\mathbf{X}) \Xi$$

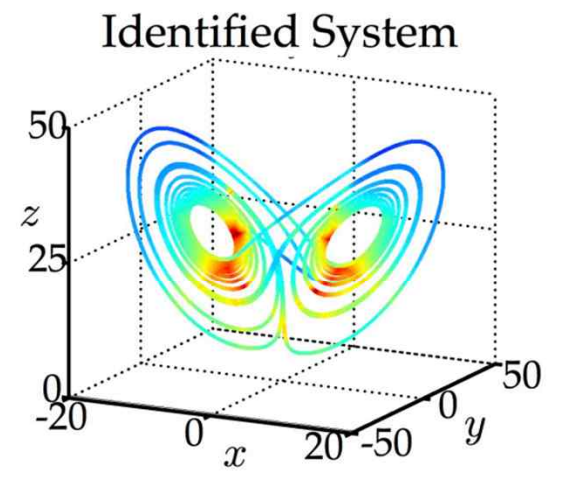
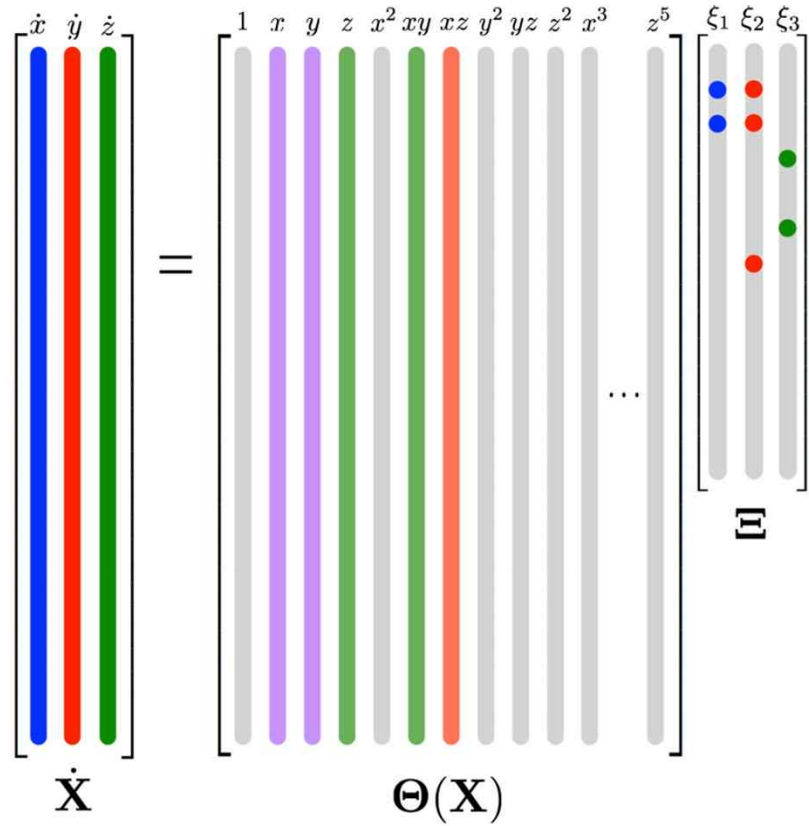
A parsimonious model will provide an accurate model fit with as few terms as possible in  $\Xi$

$$\xi_k = \arg \min_{\xi'_k} \left\| \dot{\mathbf{X}}_k - \Theta(\mathbf{X}) \xi'_k \right\|_2 + \lambda \left\| \xi'_k \right\|_1$$

SGH



$$\begin{aligned} \dot{x} &= \sigma(y - x) \\ \dot{y} &= x(\rho - z) - y \\ \dot{z} &= xy - \beta z. \end{aligned}$$



	'xi_1'	'xi_2'	'xi_3'
'x'	[-9.9996]	[27.9980]	[ 0]
'y'	[ 9.9998]	[-0.9997]	[ 0]
'z'	[ 0]	[ 0]	[-2.6665]
'xx'	[ 0]	[ 0]	[ 0]
'xy'	[ 0]	[ 0]	[ 1.0000]
'xz'	[ 0]	[-0.9999]	[ 0]
'yy'	[ 0]	[ 0]	[ 0]
...	...	...	...
'zzzzz'	[ 0]	[ 0]	[ 0]

$$\dot{\mathbf{X}} = \Theta(\mathbf{X}) \Xi$$

$$\Theta(\mathbf{X}) = \begin{bmatrix} \mathbf{1} & \mathbf{X} & \mathbf{X}^2 & \dots & \mathbf{X}^d & \dots & \sin(\mathbf{X}) & \dots \end{bmatrix}$$

# Discovering Partial Differential Equations

A major extension of SINDy modeling framework generalized the library to include partial derivative, enabling the identification of partial differential equation.

$$\Theta(\Upsilon, \mathbf{Q}) = \left[ \mathbf{1} \quad \Upsilon \quad \Upsilon^2 \quad \cdots \quad \mathcal{Q} \quad \cdots \quad \Upsilon_x \quad \Upsilon \Upsilon_x \quad \cdots \right]$$

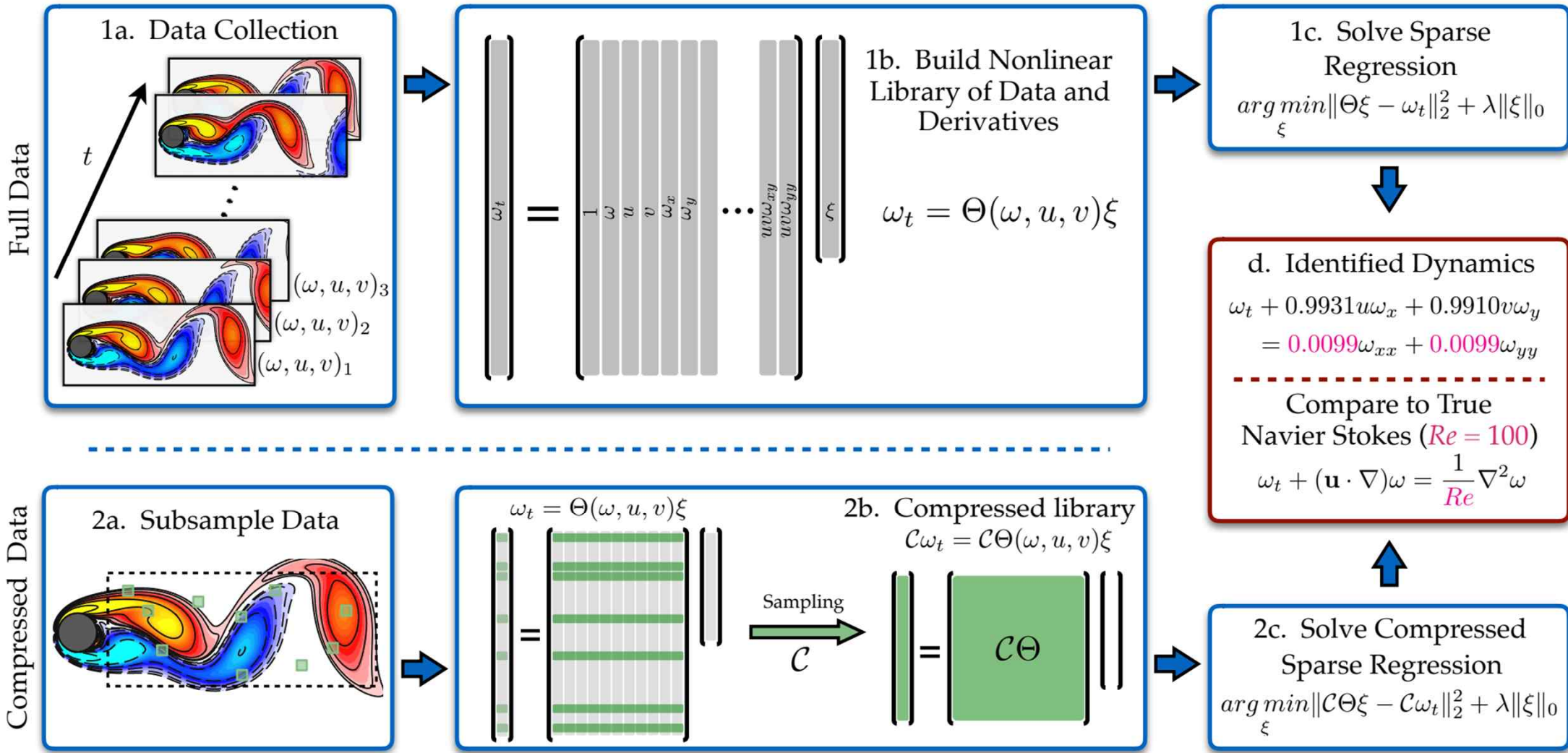
Spatial time-series



A known potential or magnitude of complex data

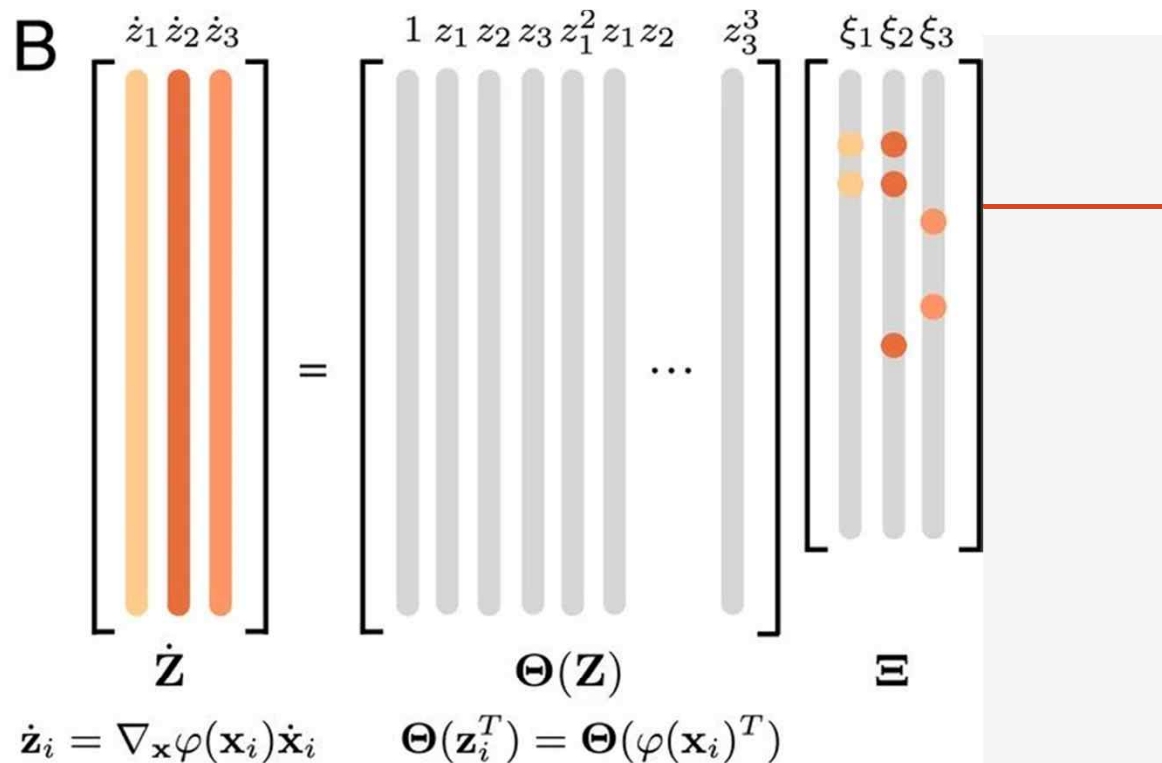
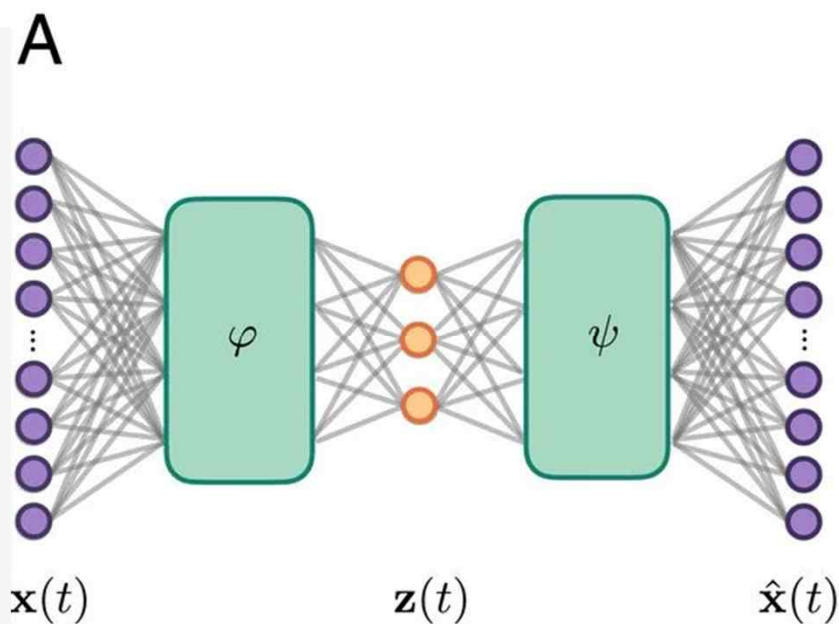
$$\Upsilon_t = \Theta(\Upsilon, \mathbf{Q}) \xi$$

$$\hat{\xi} = \arg \min_{\xi} \left\| \Theta(\Upsilon, \mathbf{Q}) \xi'_k - \Upsilon_t \right\|_2^2 + \epsilon \mathcal{K}(\Theta(\Upsilon, \mathbf{Q})) \|\xi'\|_0$$



## Data-driven discovery of coordinates and governing equations

Kathleen Champion<sup>1</sup>, Bethany Lusch<sup>2</sup>, J. Nathan Kutz<sup>1</sup>, Steven L. Brunton



$$\underbrace{\|\mathbf{x} - \psi(\mathbf{z})\|_2^2}_{\text{reconstruction loss}} + \lambda_1 \underbrace{\|\dot{\mathbf{x}} - (\nabla_{\mathbf{z}} \psi(\mathbf{z})) (\Theta(\mathbf{z}^T) \Xi)\|_2^2}_{\text{SINDy loss in } \dot{\mathbf{x}}} + \lambda_2 \underbrace{\|(\nabla_{\mathbf{x}} \mathbf{z}) \dot{\mathbf{x}} - \Theta(\mathbf{z}^T) \Xi\|_2^2}_{\text{SINDy loss in } \dot{\mathbf{z}}} + \lambda_3 \underbrace{\|\Xi\|_1}_{\text{SINDy regularization}}$$

Schematic of the SINDy autoencoder method for simultaneous discovery of coordinates and parsimonious dynamics

# Application to structural vibration

The application of Dynamic Mode Decomposition (DMD) to the extraction of modal properties of linear mechanical systems, i.e., experimental modal analysis (EMA).

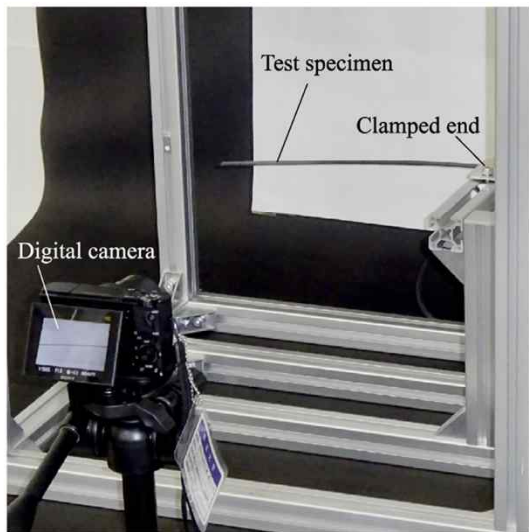
Data-driven experimental modal analysis by  
Dynamic Mode Decomposition  
Akira Saito\*, Tomohiro Kuno

<https://doi.org/10.1016/j.jsv.2020.115434> 0022-460X/©2020 Elsevier Ltd. All rights reserved.

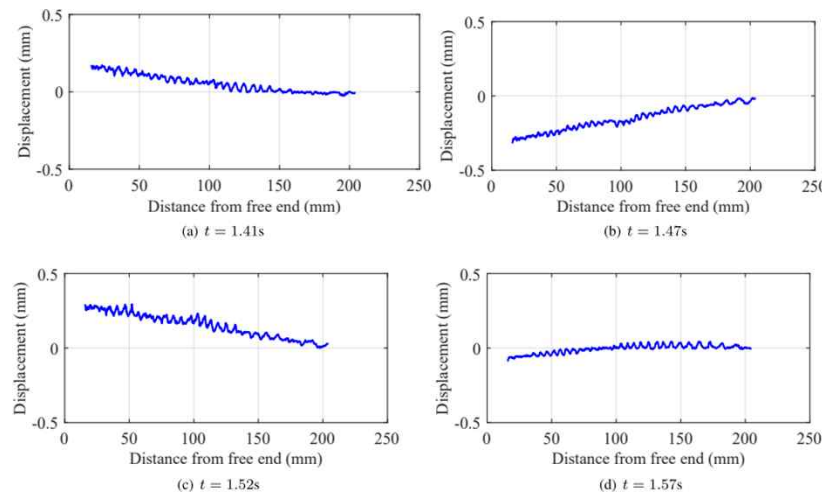
Journal of Sound and Vibration



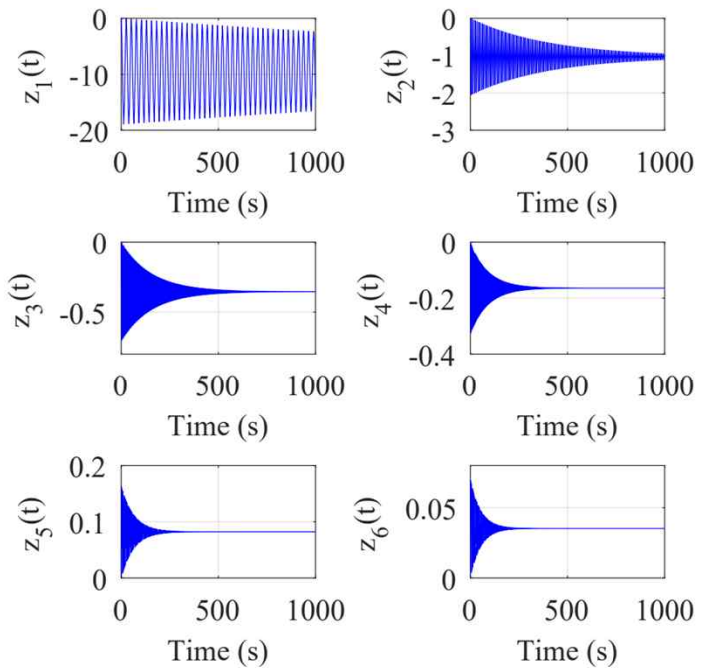
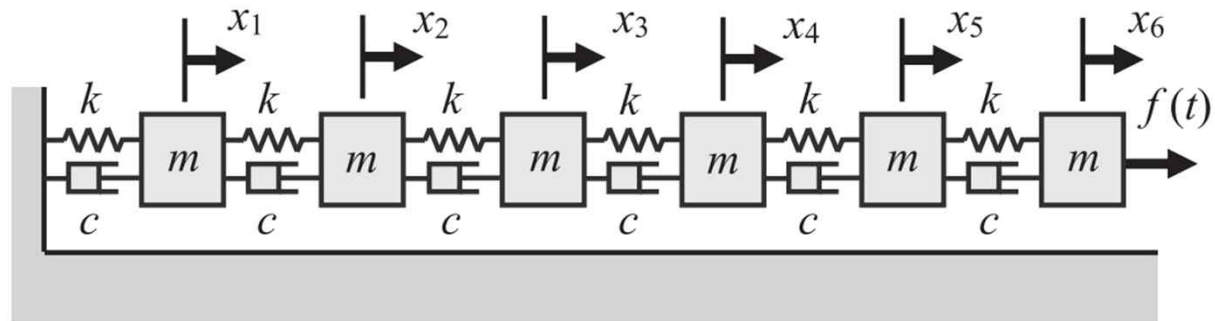
(a) Test specimen



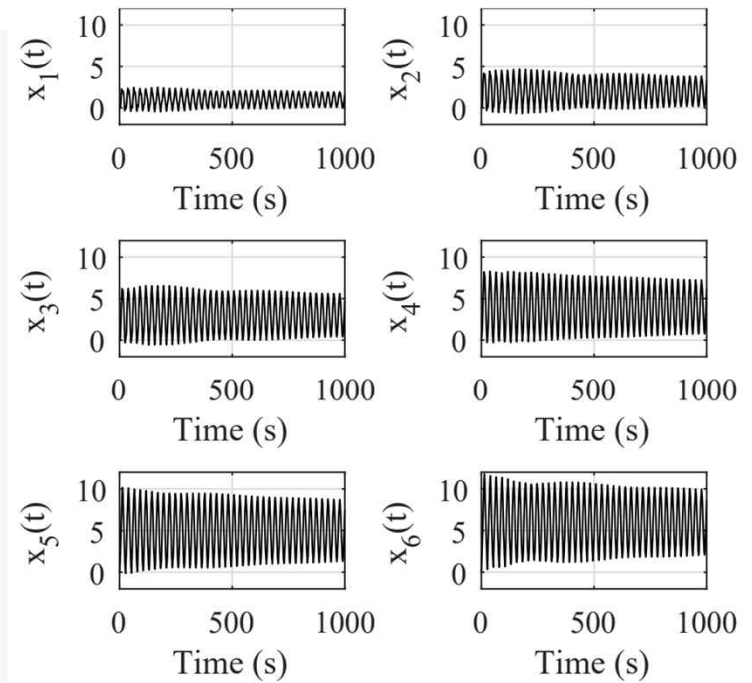
(b) Experimental setup







(a) Time response of modal coordinates ( $\mathbf{z}(t)$ )



(b) Time response of physical DOF ( $\mathbf{x}(t)$ )

# Application of SINdy to Structural Dynamics

Sparse structural system identification method for nonlinear dynamic systems with hysteresis/inelastic behavior

By Zhilu Lai, Satish Nagarajaiah

Mechanical Systems and Signal Processing 117 (2019) 813–842

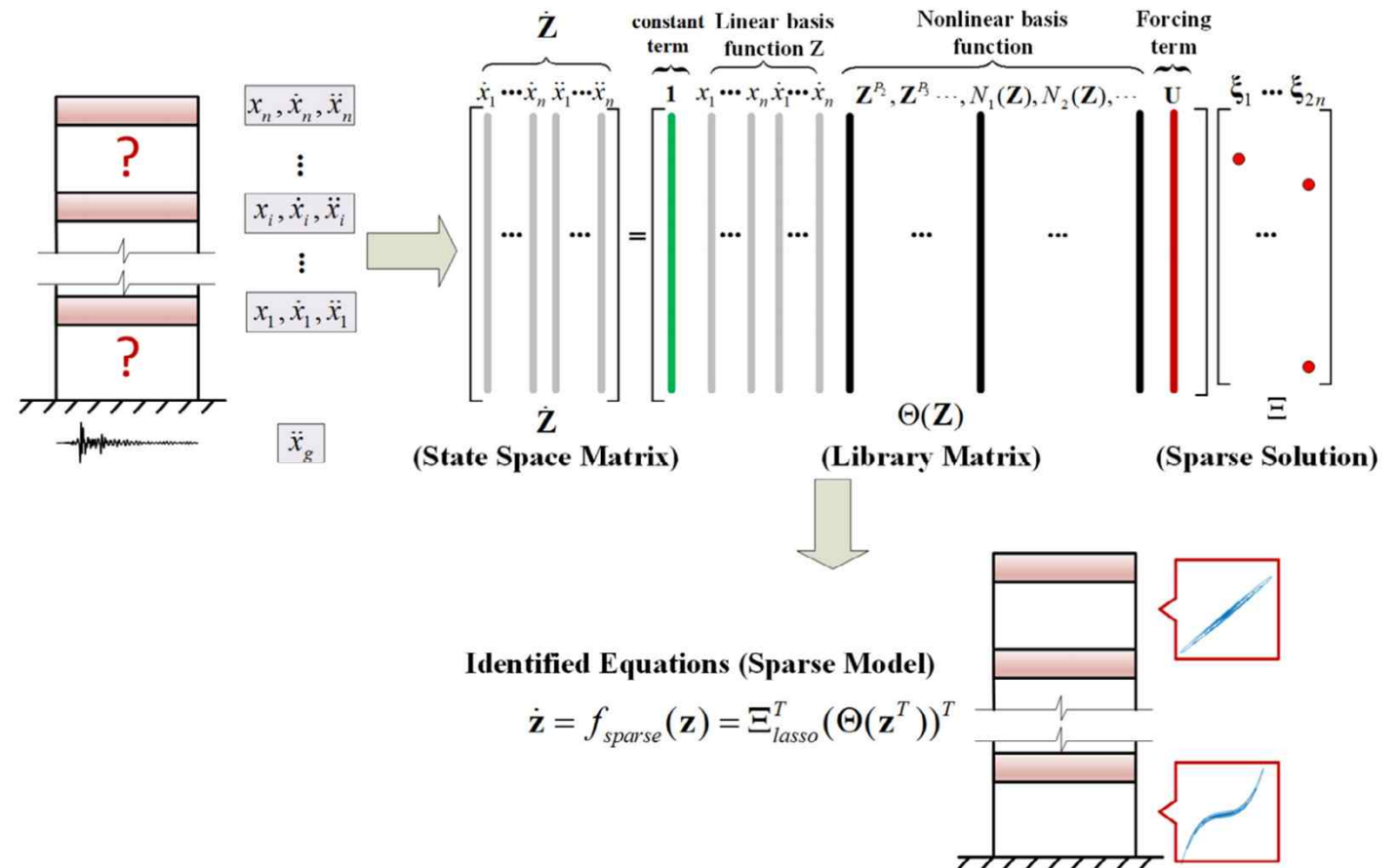


Fig. 2. Schematic procedure of sparse identification to nonlinear elastic structural systems.

# Steps in SINDy for Dynamics

---

## 1. Data Preparation

Obtain the measured data  $x_i(t)$ ,  $\ddot{x}_i(t)$  and  $\ddot{x}_g(t)$  at time  $t_1, t_2, \dots, t_m$  and compute  $\dot{x}_i(t)$ , and  $\dot{a}_{h,i}(t)$

## 2. Assemble $\mathbf{Z}_a$ and $\mathbf{Z}_h$

Depending on the choice of types of nonlinearities (nonlinear elastic or nonlinear inelastic),  $\mathbf{Z}_a$  and  $\mathbf{Z}_h$  are constructed through Eq. (30) and Eq. (32) for nonlinear inelastic behavior (or Eq. (4) for nonlinear elastic).

## 3. Construct Library Matrix

Library matrix is constructed with only measured data, according to the choice of types of basis functions used to represent different nonlinearities as in Eq. (35) for nonlinear inelastic or Eq. (6) for nonlinear elastic.

## 4. Sparse Feature/Model Extraction (training data)

With a certain regularized parameter  $\lambda$ , solve the  $\ell_1$  regularized regression problem (Eq. (26) for nonlinear inelastic or Eq. (8) for nonlinear elastic) by LASSO to get sparse model parameters and its corresponding AIC value.

## 5. Model Selection

Repeat step 2 to step 4 using various values of  $\lambda$  and form an AIC curve. Pick the optimal  $\lambda$  where the AIC curve has significant slope change.

**6. Sparse Feature/Model Evaluation (testing data)** Testing ground motion as input to the sparse model to verify the generalization ability of the model. Note: testing data is not used in training as dictated by cross validation.

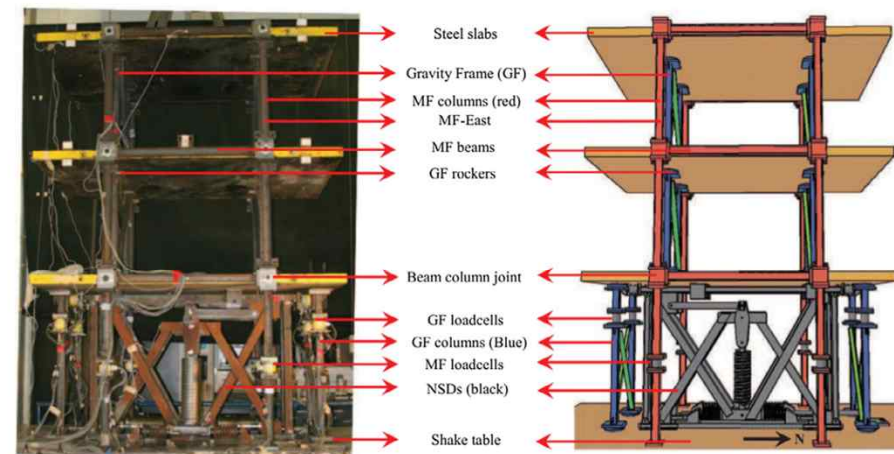
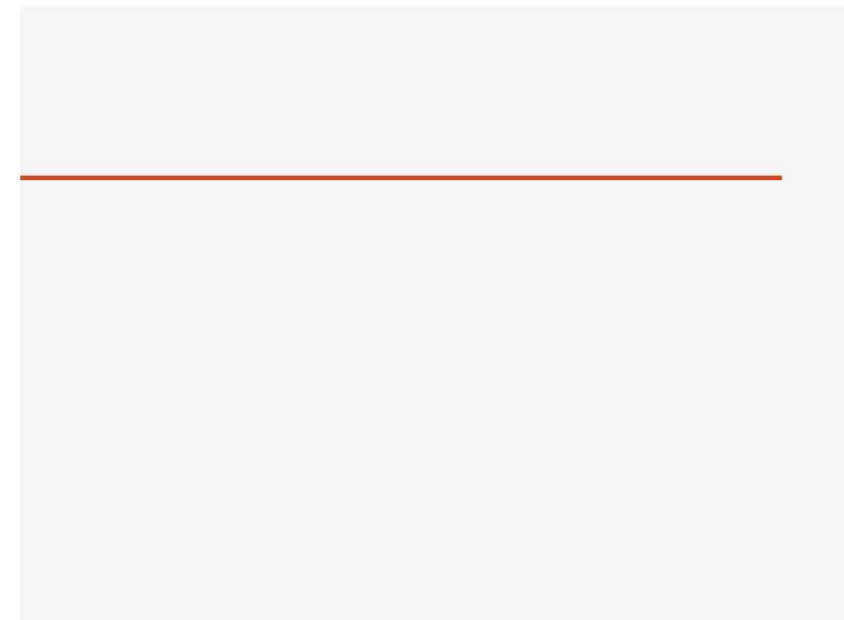
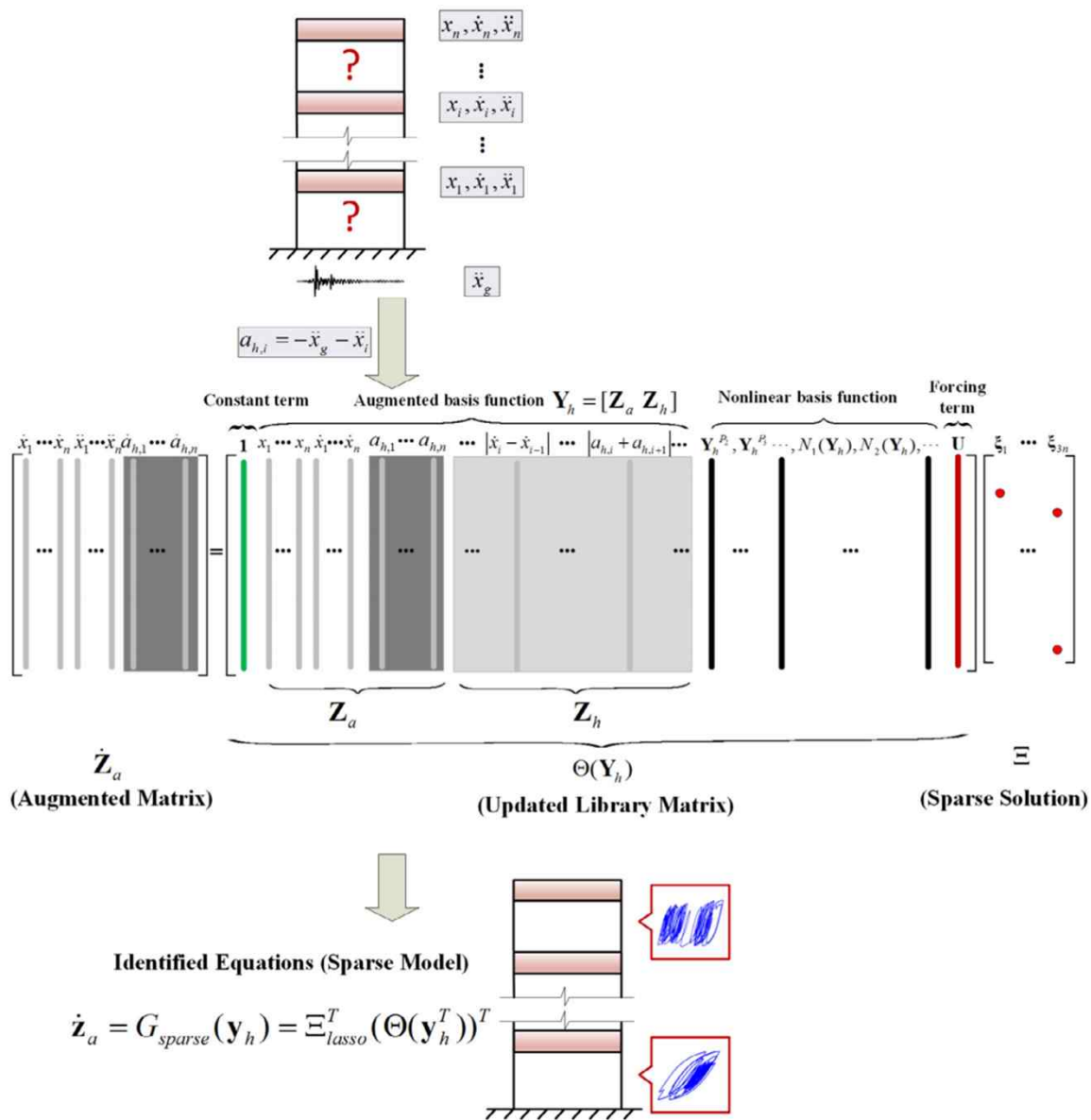
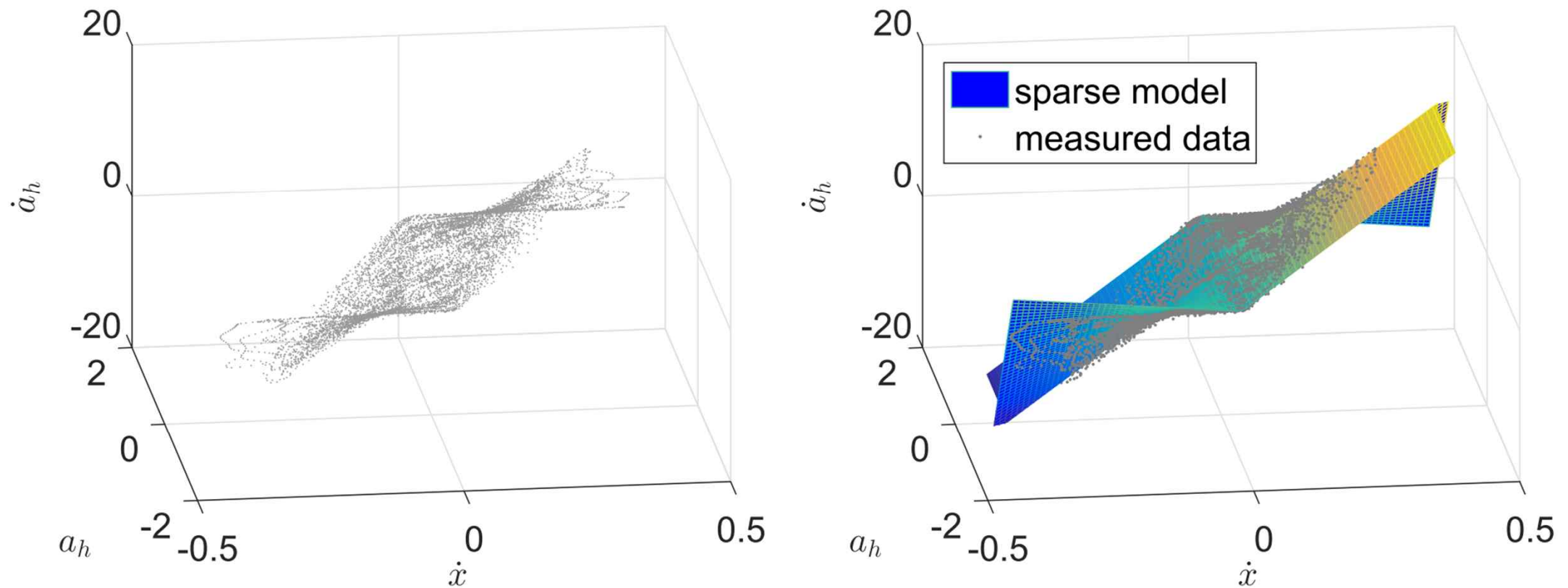


Fig. 18. Shake table testing of a 3-story structure with a NSD installed in the first floor.

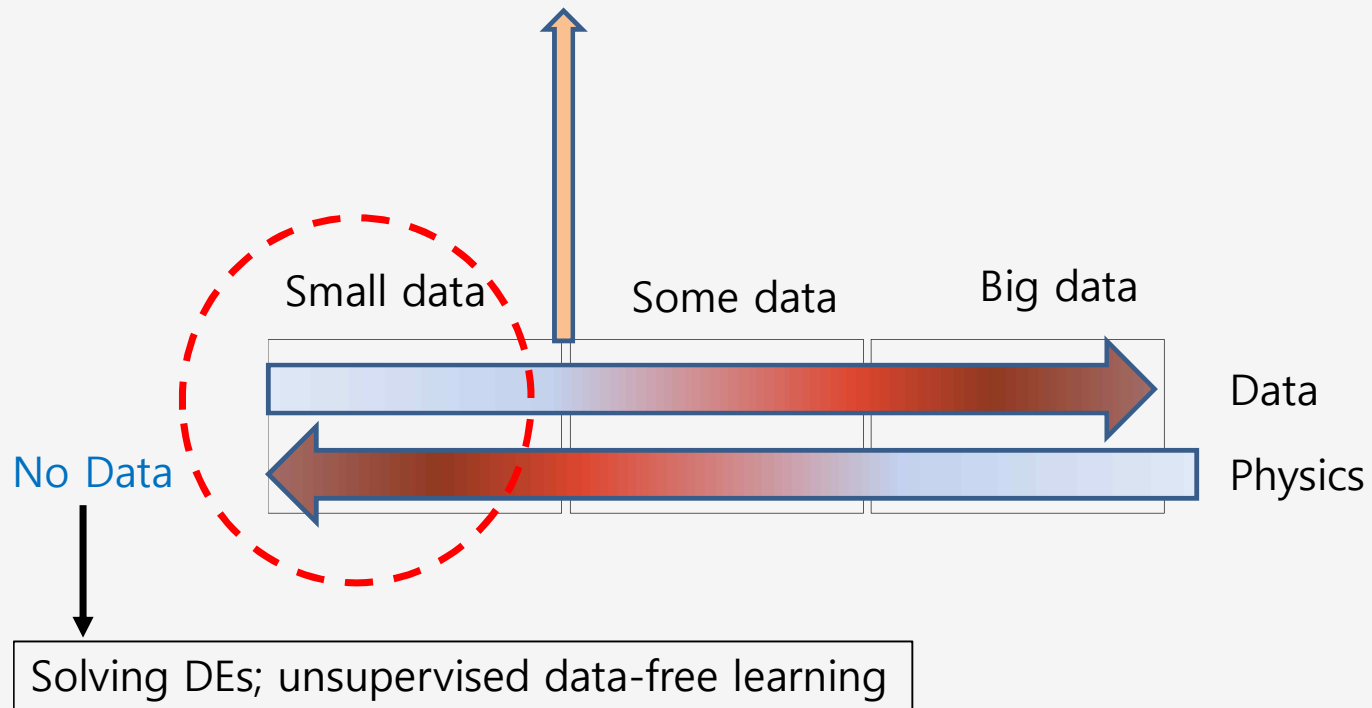
Fig. 3. Schematic procedure of proposed augmented sparse identification for nonlinear inelastic structural systems with hysteresis and permanent deformation.

# Comparison



**Fig. 11.** measured data (left); comparison between measured data and function surface identified in the sparse model (right).

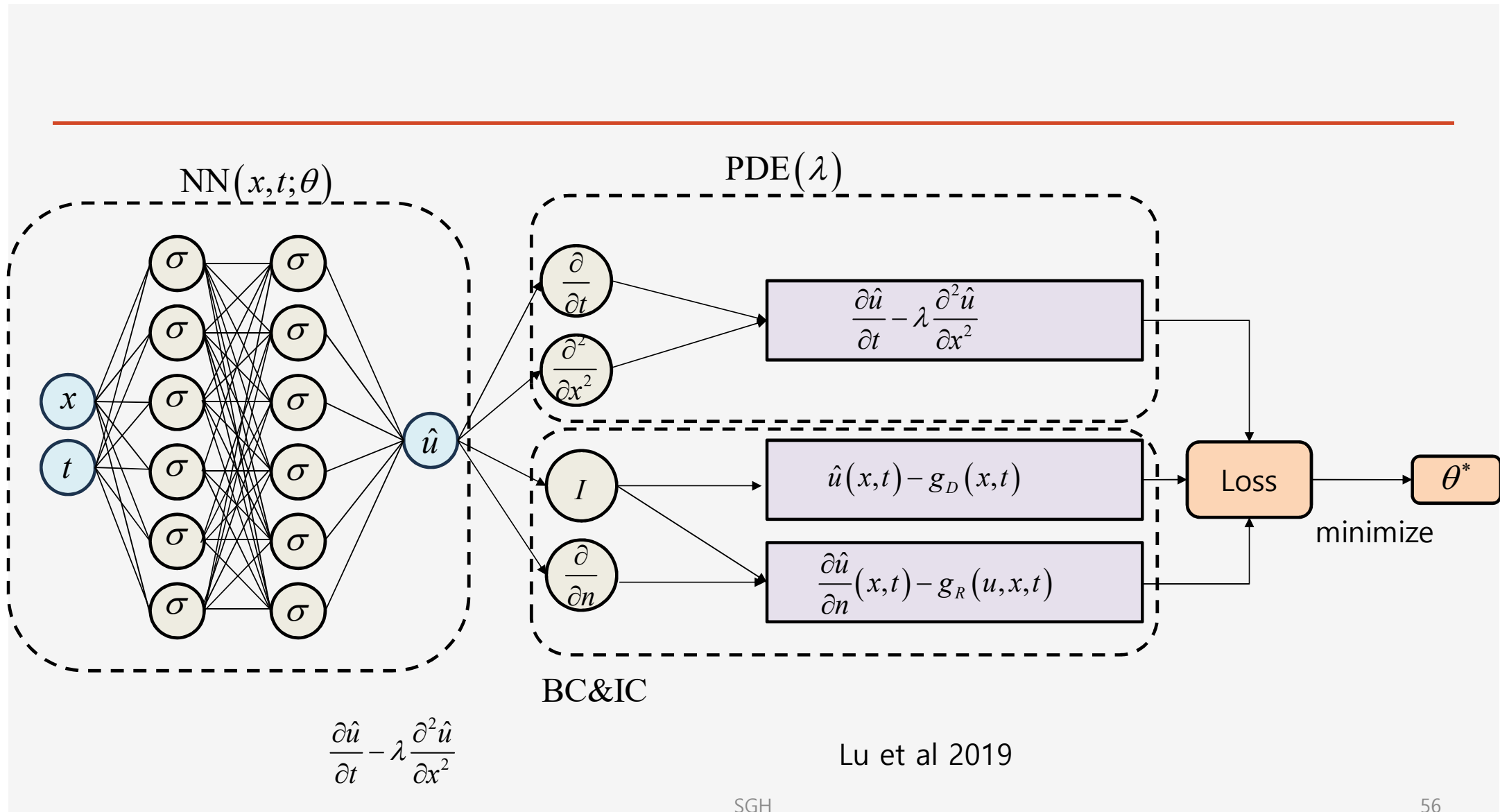
# Physics informed Neural Network



# Why SciML

---

- Scientific machine learning is an emerging discipline within the data science community. SciML seeks to address domain-specific data challenges and extract insights from scientific data sets through innovative methodological solution.
- SciML draws on tools from both machine learning and scientific computing to develop new methods for scalable, domain-aware, robust, reliable, and interpretable learning and data analysis, and will be critical in driving **the next wave** of **data-driven scientific discovery** in the physical and engineering sciences.





# Behavior of system and Optimization objective

Partial differential equation:  $F(u; \theta)(\mathbf{x}) = 0 \longrightarrow F(u; \theta)(\mathbf{x}) = F(u, \mathbf{x}; \theta) = 0, \quad x \in \Omega$

Initial conditions:  $I(u; \theta)(x, t_0) = 0$

Boundary conditions:  $B(u; \theta)(x, t) = 0$

Optimization objective:  $\min_{f \in H} L(f; D) + \Omega(g)$

$$L = \underbrace{\frac{\lambda_r}{N_r} \sum_{i=1}^{N_r} \|F(u_w; \theta)(\mathbf{x}_i)\|^2}_{\text{Residual loss}} + \underbrace{\frac{\lambda_i}{N_i} \sum_{i=1}^{N_i} \|I(u_w; \theta)(\mathbf{x}_i)\|^2}_{\text{Initial condition}}$$

$$+ \underbrace{\frac{\lambda_b}{N_b} \sum_{i=1}^{N_b} \|B(u_w; \theta)(\mathbf{x}_i)\|^2}_{\text{Boundary condition}} + \underbrace{\frac{\lambda_d}{N_d} \sum_{i=1}^{N_d} \|u_w(\mathbf{x}_i) - u(\mathbf{x}_i)\|^2}_{\text{Regular data loss}}$$

# Loss Reweighting

$$\hat{\lambda}_i = \frac{\max \{ \nabla_w L_r(w_n) \}}{|\nabla_w L_r(w_n)|}$$

$$K = \begin{pmatrix} K_{bb} & K_{br} \\ K_{br} & K_{rr} \end{pmatrix}$$

$$(K_{bb})_{i,j} = \left\langle \frac{du_w(\mathbf{x}_i^b)}{dw}, \frac{du_w(\mathbf{x}_j^b)}{dw} \right\rangle$$

$$(K_{bb})_{i,j} = \left\langle \frac{du_w(\mathbf{x}_i^b)}{dw}, \frac{dF(u_w)(\mathbf{x}_j)}{dw} \right\rangle$$

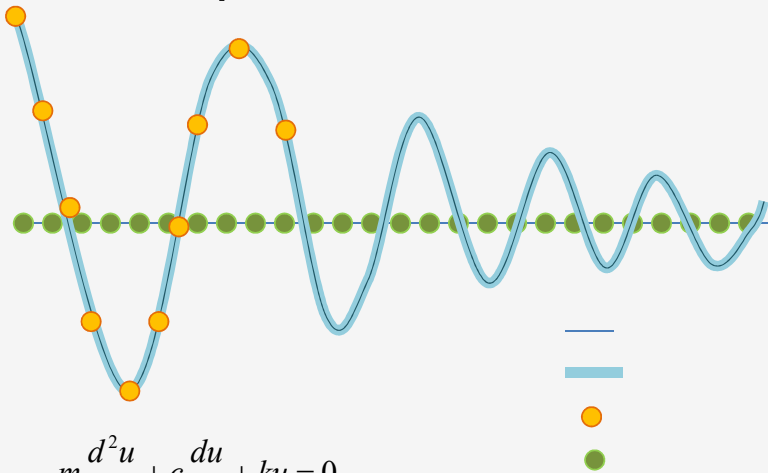
$$(K_{bb})_{i,j} = \left\langle \frac{dF(u_w)(\mathbf{x}_i)}{dw}, \frac{dF(u_w)(\mathbf{x}_j)}{dw} \right\rangle$$

$$\lambda_b = \frac{\text{Tr}(K)}{\text{Tr}(K_{bb})}$$

$$\lambda_r = \frac{\text{Tr}(K)}{\text{Tr}(K_{rr})}$$

# Advantage and limitation

## • Example



$$m \frac{d^2 u}{dt^2} + c \frac{du}{dt} + ku = 0$$

$$NN(t; \theta) \approx u(t)$$

$$L(\theta) = \frac{1}{N} \sum_i^N (\text{Boundary loss}) (NN(t_i; \theta) - u(t_i))^2$$

$$+ \frac{\lambda}{M} \sum_j^M \left( \left[ m \frac{d^2}{dt^2} + c \frac{d}{dt} + k \right] NN(t_j; \theta) \right)^2$$

Physics loss

## Advantages

- Mesh-free
- Can jointly solve forward and inverse problems
- Mostly unsupervised
- Can perform well for high-dimensional PDEs

## Limitation

- Computational cost often high
- Can be hard to optimize
- Challenging to scale to high-frequency, multi-scale problems

from ETH 401-4656-21L Deep learning in Scientific computing 2023

# Neural operator

## DeepONets

Physical Informed DeepONets

Graph Operator Networks

## Fourier Operator Networks

PINO

## Caltech group

Nikola Kovachki

Zongyi Li

Anima Anandkumar

Andrew Stuart

.....

## Brown University

Lu Lu

Pengzhan Jin

George Em

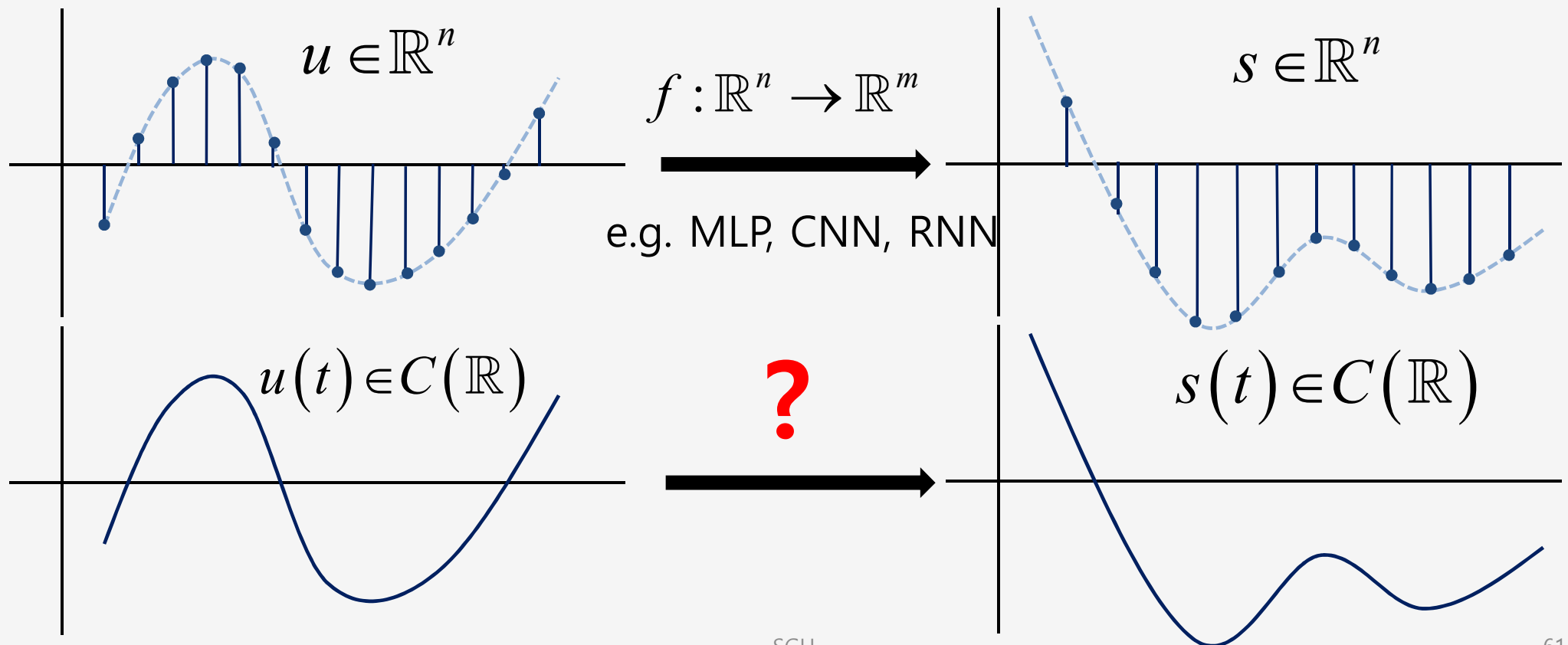
Karniadakis

**Not vector-to-vector mapping**

**Function-to-function mapping**

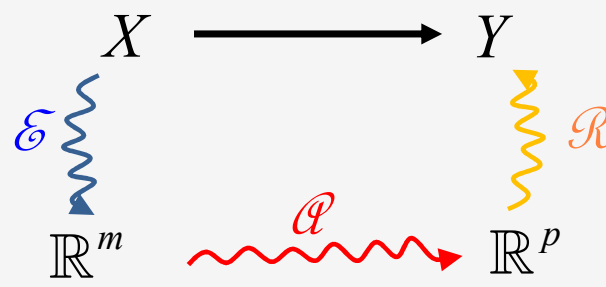
# How to learn function space

Take **discrete measurement** of functional data and use standard ML models on the finite dimensional discretization



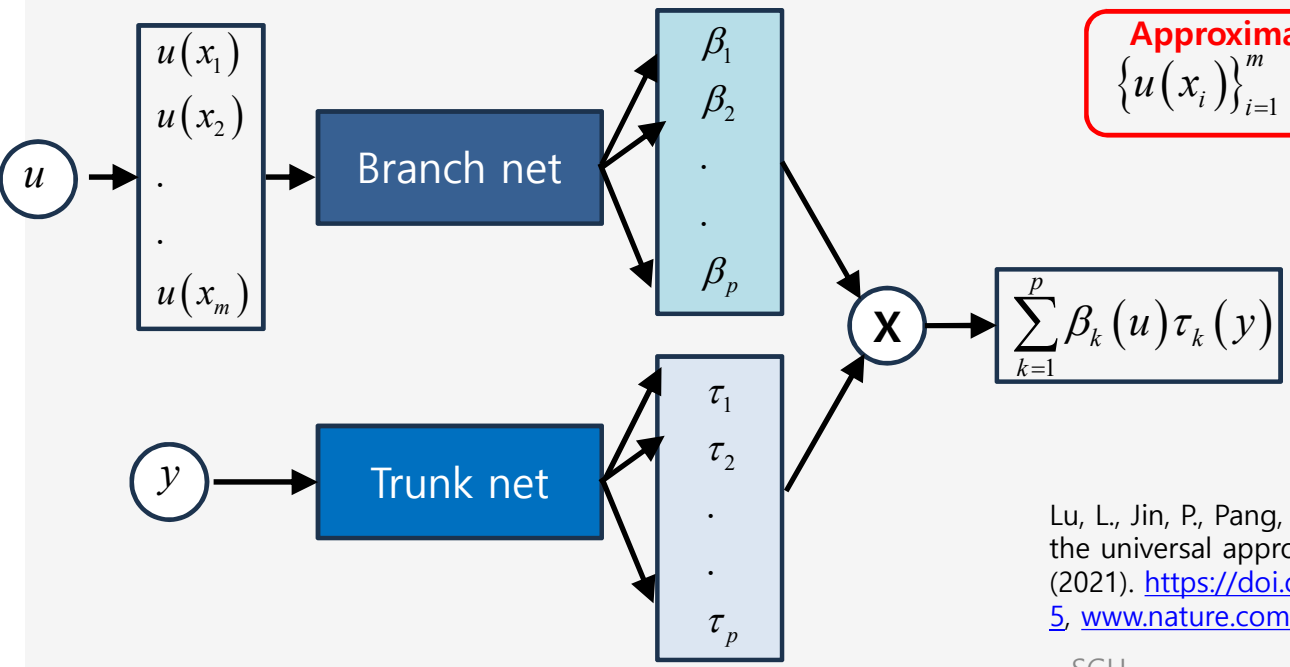
# DeepOperatorNet

**Encoding**  
 $u \mapsto \{u(x_i)\}_{i=1}^m$



**Reconstruction**  
 $\{\beta_k(u)\}_{k=1}^p \mapsto \sum_{k=1}^p \beta_k(u) \tau_k(y)$

**Approximation**  
 $\{u(x_i)\}_{i=1}^m \mapsto \{\beta_k(u)\}_{k=1}^p$



Lu, L., Jin, P., Pang, G., et al.: Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. Nat. Mac. Intell. 3(3), 218–229 (2021). <https://doi.org/10.1038/s42256-021-00302-5>, [www.nature.com/articles/s42256-021-00302-5](http://www.nature.com/articles/s42256-021-00302-5)



# Neural operator

---

$$u = Q(K_l \circ \sigma_l \circ \cdots \circ \sigma_1 \circ K_0) P v$$

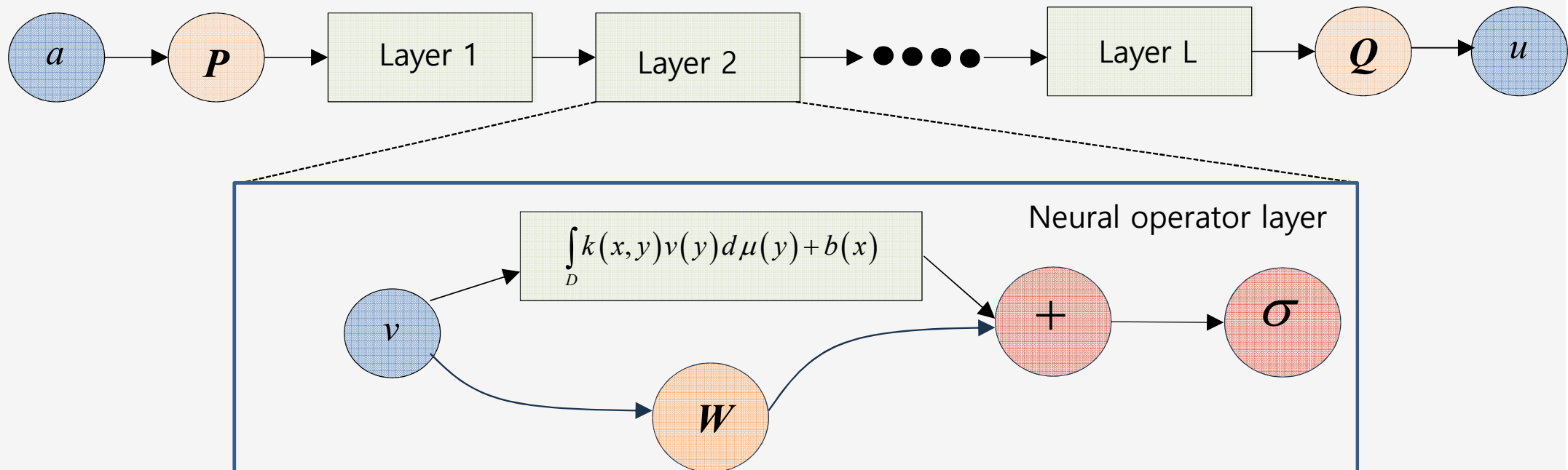
P, Q are local network (encoder, decoder)

P lifts the input to a high dimensional channel space.

Q projects the representation back to the original space.

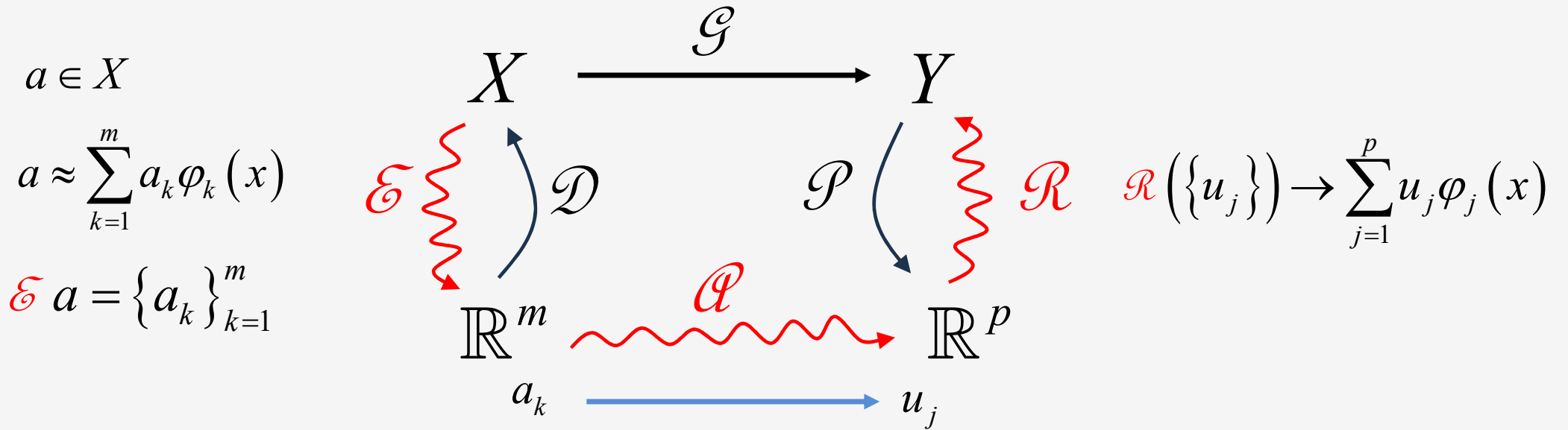


# Neural operator architecture schematic



The input function  $a$  is passed to a pointwise lifting operator  $P$  that is followed by  $T$  layers of integral operators and pointwise non-linearity operator  $\sigma$ . In the end, the pointwise projection operator  $Q$  outputs the function  $u$ .

# Operator Learn Architecture



Architecture	Encoder	Approximator	Reconstructor
Spectral Neural Operator	Coeffs	DNNs	Fourier/Chebyshev basis

# 2D steady Darcy flow equation

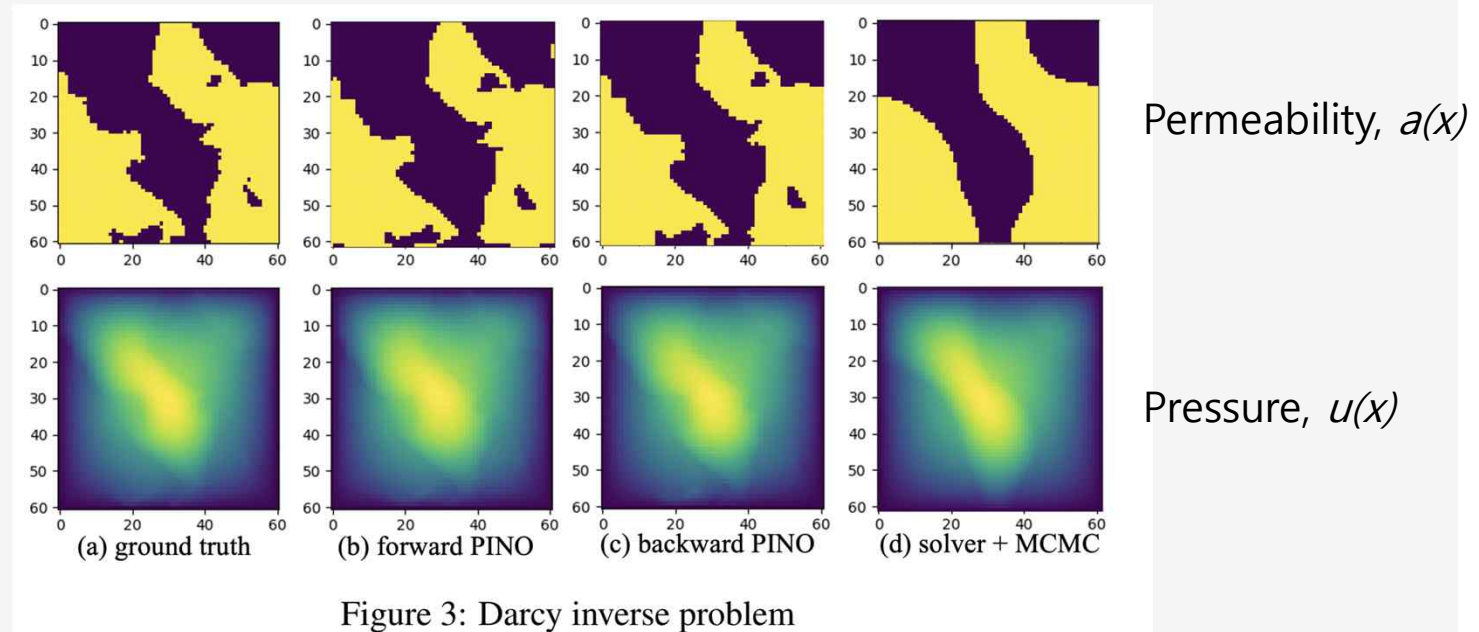
$$-\nabla \cdot (a(x) \nabla u(x)) = f(x) \quad x \in (0,1)^2$$

$$u(x) = 0 \quad x \in \partial(0,1)^2$$

PHYSICS-INFORMED  
NEURAL OPERATOR  
FOR LEARNING PARTIAL  
DIFFERENTIAL  
EQUATIONS

By Anonymous authors

Under review as a conference paper  
at ICLR 2022



# Fourier Layer

---

Use convolution as integral operator  
and implement with Fourier transform

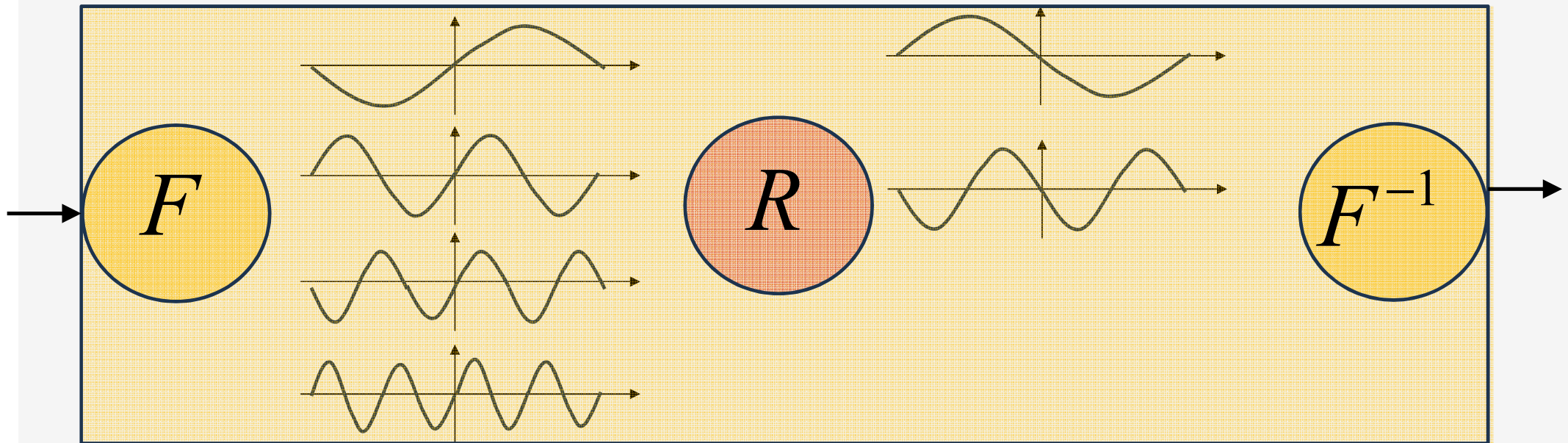
$$(K(\alpha; \phi)v_t) := \int_D k(x, y, a(x), a(y); \phi) v_t(y) dy$$

$$(K(\phi)v_t) = F^{-1}(R_\phi \cdot (Fv_t))(x)$$

# Fourier layer

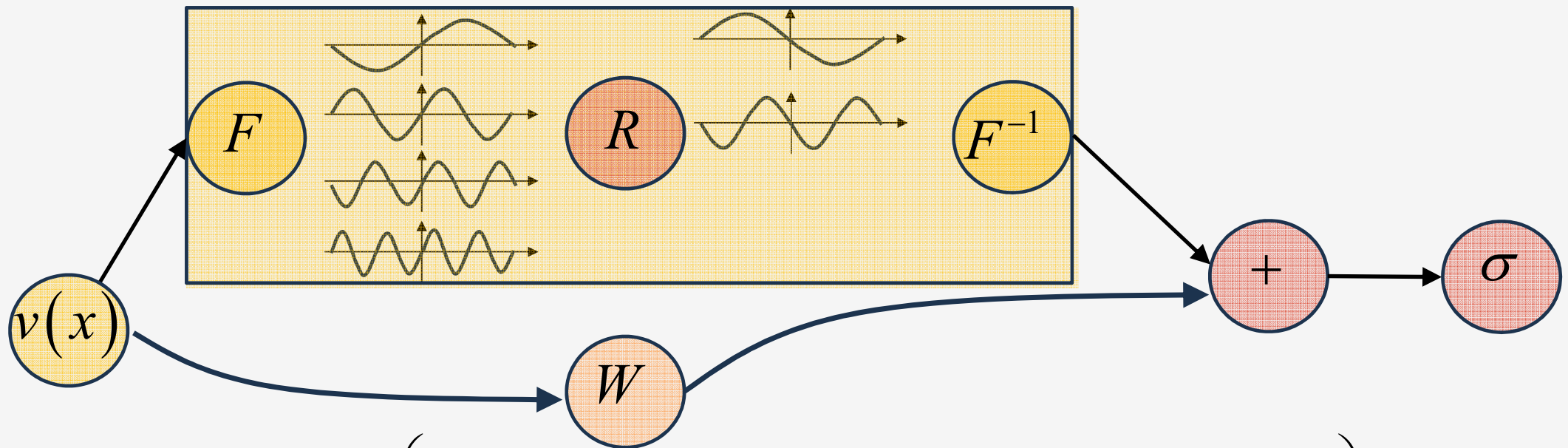
1. Fourier transform
2. Linear transform
3. Inverse Fourier transform

$$(K(\phi)v_t) = F^{-1}(R_\phi \cdot (Fv_t))(x)$$



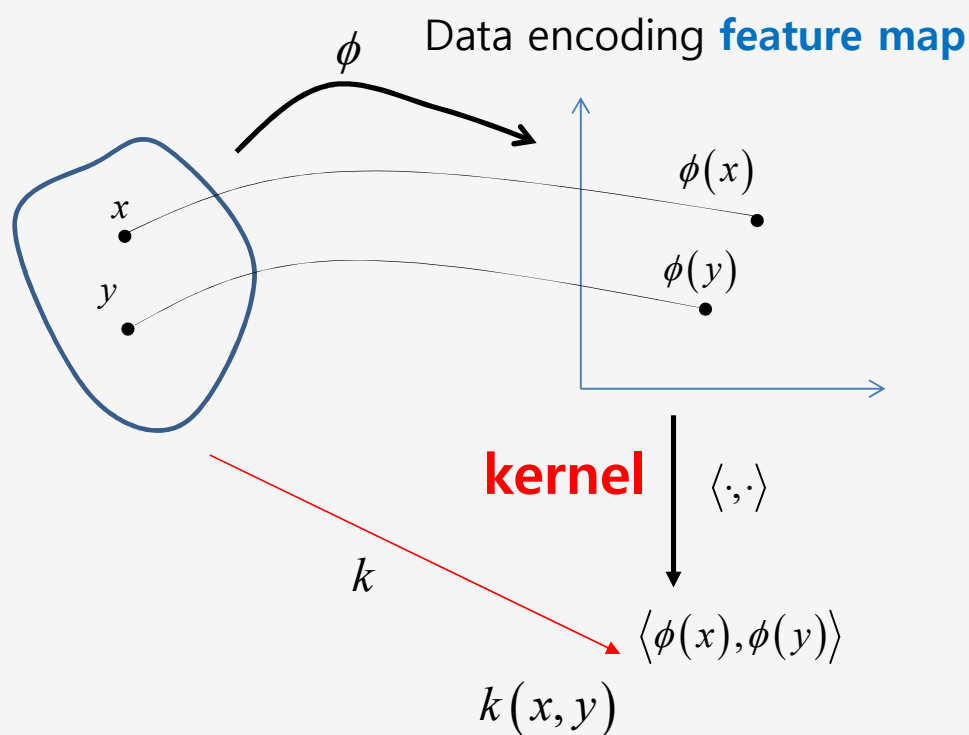
# Fourier layer

The linear transform  $W$  outside keep the track of the location information ( $x$ ) and non-periodic boundary



$$v_{t+1}(x) = \sigma \left( Wv_t(x) + \int_D k_\phi(x, y, a(x), a(y)) v_t(y) v_x(dy) \right)$$

# Kernelization



## Polynomial kernel example

$$\phi: \mathbb{R}^2 \Rightarrow \mathbb{R}^3$$

$$\phi: x_1, x_2 \rightarrow z_1, z_2, z_3$$

$$\text{where } z_1 = \sqrt{2}x_1x_2 \quad z_2 = x_1^2 \quad z_3 = x_2^2$$


$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle$$

$$\begin{aligned} k(\mathbf{x}, \mathbf{z}) &= (\mathbf{x}^T \mathbf{z})^2 = (x_1z_1 + x_2z_2)^2 \\ &= x_1^2z_1^2 + 2x_1z_1x_2z_2 + x_2^2z_2^2 \\ &= (x_1^2, \sqrt{2}x_1x_2, x_2^2) \begin{pmatrix} z_1^2 \\ \sqrt{2}z_1z_2 \\ z_2^2 \end{pmatrix} \\ &= \phi(\mathbf{x})^T \phi(\mathbf{z}) \end{aligned}$$

# Kernel Trick

---

- Computation in explicit, **high-dimensional feature maps are expensive**
- For some feature maps, we can, however, compute distances between point cheaply
  - Without explicitly constructing the high-dimensional space at all

• Example: quadratic feature map for  $\mathbf{x} = (x_1, \dots, x_p)$  

$p$ \_dimension  $\Rightarrow$   $3p$ \_dimension

$$\Phi(\mathbf{x}) = (x_1, \dots, x_p, x_1^2, \dots, x_p^2, \sqrt{2}x_1x_2, \dots, \sqrt{2}x_{p-1}x_p)$$

- **A kernel function exists** for this feature map to compute dot products

$$k_{quad}(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j + (\mathbf{x}_i \cdot \mathbf{x}_j)^2$$


- **Skip computation of**  $\Phi(\mathbf{x}_i)$  and  $\Phi(\mathbf{x}_j)$  and compute  $k(\mathbf{x}_i, \mathbf{x}_j)$  **directly**



---

Kernel Function  $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$

"Gram Matrix"

$$K = \begin{bmatrix} \phi(\mathbf{x}_1)^T \phi(\mathbf{x}_1) & \phi(\mathbf{x}_1)^T \phi(\mathbf{x}_2) & \cdot & \phi(\mathbf{x}_1)^T \phi(\mathbf{x}_N) \\ \phi(\mathbf{x}_2)^T \phi(\mathbf{x}_1) & \phi(\mathbf{x}_2)^T \phi(\mathbf{x}_2) & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \phi(\mathbf{x}_N)^T \phi(\mathbf{x}_1) & \phi(\mathbf{x}_N)^T \phi(\mathbf{x}_2) & \cdot & \phi(\mathbf{x}_N)^T \phi(\mathbf{x}_N) \end{bmatrix}$$

$$= \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \cdot & k(\mathbf{x}_1, \mathbf{x}_N) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ k(\mathbf{x}_N, \mathbf{x}_1) & k(\mathbf{x}_N, \mathbf{x}_2) & \cdot & k(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix}$$

---


$$f(x) = \sum_t w_t \underbrace{\phi(\mathbf{x}_t) \cdot \phi(\mathbf{x})}_{k(\mathbf{x}_t, \mathbf{x})}$$

training data      New data      New data  
 (bracketed under  $\phi(\mathbf{x}_t)$ )      (bracketed under  $\phi(\mathbf{x})$ )

$$\Psi_{GPS}(\mathbf{x}; p, \theta) = \exp\left(\sum_t^{N_t} w_t \underbrace{k_p^\theta(\mathbf{x}, \mathbf{x}_t)}_{\text{kernel Training data}}\right)$$

(bracketed under  $\sum_t^{N_t}$ )      (bracketed under  $k_p^\theta(\mathbf{x}, \mathbf{x}_t)$ )

Exponential ensures **product** separable

# Noiseless GP regression

We observe a training set  $D = \{(\mathbf{x}_i, \mathbf{f}_i), i = 1:N\}$  where  $f_i = f(\mathbf{x}_i)$

Given a test set  $X^*$  of size  $N^* \times D$ , we want to **predict the function output**  $\mathbf{f}^*$ .

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim N\left(\begin{pmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu}_* \end{pmatrix}, \begin{bmatrix} K(X, X) & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right)$$

$K(X, X): N \times N$   
 $K(X, X_*): N \times N_*$   
 $K(X_*, X_*): N_* \times N_*$

$$k(x, x') = \sigma_f^2 \exp\left(-\frac{1}{2l^2}(x - x')^2\right)$$

$$p(\mathbf{f}_* | \mathbf{X}_*, \mathbf{X}, \mathbf{f}) = N(\mathbf{f}_* | \boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*)$$

$$\boldsymbol{\mu}_* = \boldsymbol{\mu}(\mathbf{X}_*) + \mathbf{K}_*^T \mathbf{K}^{-1} (\mathbf{f} - \boldsymbol{\mu}(\mathbf{X}))$$

$$\boldsymbol{\Sigma}_* = \mathbf{K}_{**} - \mathbf{K}_*^T \mathbf{K}^{-1} \mathbf{K}_*$$

# Noisy GP regression

---

$$y = f(X) + \varepsilon, \quad \text{where } \varepsilon \sim N(0, \sigma_y^2)$$

$$p(\mathbf{y}|\mathbf{X}) = \int p(\mathbf{y}|\mathbf{f}, \mathbf{X})p(\mathbf{f}|\mathbf{X})d\mathbf{f}$$

$$p(\mathbf{f}|\mathbf{X}) = N(\mathbf{f}|\mathbf{0}, \mathbf{K})$$

$$p(\mathbf{y}|\mathbf{f}) = \prod_i N(y_i|f_i, \sigma_y^2)$$

$$\text{cov}[\mathbf{y}|\mathbf{X}] = \mathbf{K} + \sigma_y^2 \mathbf{I}_N \triangleq \mathbf{K}_y$$

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim N\left(\mathbf{0}, \begin{bmatrix} K_y & K_* \\ K_*^T & K_{**} \end{bmatrix}\right)$$

$$p(\mathbf{f}_*|\mathbf{X}_*, \mathbf{X}, \mathbf{f}) = N(\mathbf{f}_*|\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*)$$

$$\boldsymbol{\mu}_* = \mathbf{K}_*^T \mathbf{K}^{-1} \mathbf{y}$$

$$\boldsymbol{\Sigma}_* = \mathbf{K}_{**} - \mathbf{K}_*^T \mathbf{K}^{-1} \mathbf{K}_*$$

## Concluding remark: Takeaway (by Zongyi Li @caltech)

1. Data-driven ML: learn the equation
2. Neural operator-learning: parametrize the mesh-invariant operator
3. Fourier method: efficient for continuous inputs and outputs
4. SciML: accurate than other deep learning method, faster than conventional solvers
5. Future: scale up for engineering applications

# Education and conference

Join Us Events News

ETH zürich

Video portal Campus Events Lectures Speakers Support

## Deep Learning and Scientific Computing

AMCS | CEMSE | STAT  
Graduate Seminar  
Event Start 2022-09-27 - 12:00  
Event End 2022-09-27 - 13:00  
Location Building 9, level 2, Room 2322  
deep learning scientific computing

Jinchao Xu, Professor of Applied Mathematics and Computational Sciences, CEMSE, KAUST  
Tuesday, September 27, 12:00 - 13:00  
Building 9, Level 2, Room 2322

Jinchao Xu Professor, Applied Mathematics and Computational Sciences

Abstract

Deep Learning has found many successful applications in artificial intelligence (AI) for tasks such as image recognition, natural language processing, and autonomous driving. In this talk, I will first give an elementary introduction to basic deep learning models and training algorithms from a scientific computing viewpoint. Using image classification as an example, I will try to give mathematical explanations of why and how some popular deep learning models such as convolutional neural network (CNN) work. Most of the talk will be assessable to an audience who have basic knowledge of calculus and matrix. Toward the end of the talk, I will touch upon some advanced topics to demonstrate the potential of new mathematical insights for helping understand and improve the efficiency of deep learning technology.

Events

- 24 Mar CEMSE Public Colloquium: Secure Power Systems in the Age of Renewable... Charalambos Konstantinou, 12:00-13:30, Building 9, Level 2, Room 2325
- 24 Mar CEMSE Graduate Seminar: Canceled: Printed Electronics for... Dr. Mohammad Vaseem and Dr. Sabandar Raaf..., 12:00-13:00, Building 9, Level 2, Room 2325
- 24 Mar CS PhD Dissertation Defense: Towards Trustworthy AI...

Lectures

- D-ARCH
- D-BAUG
- D-BIOL
- D-BSSE
- D-CHAB
- D-ERDW
- D-GESS
- D-HEST
- D-INFK
- D-ITET
- D-MATH 2024
- 2023

Deep Learning in Scientific Computing

Search all videos

Deep Learning in Scientific Computing  
Introduction to Differentiable Physics - Part 2  
Spring Semester 2023  
Siddhartha Mishra  
Bun Moseley  
ETH zürich

Welcome

## International Conference on Scientific Computing and Machine Learning 2024

Kyoto & Online, Japan, March 19 – 23, 2024.

### Ph.D. Course on Scientific Machine Learning



We offer a Ph.D. course on Scientific Machine Learning.

The course is offered with support from the DTU Compute Graduate School (ITMAN) and the Danish Center for Applied Mathematics and Mechanics (DCAMM) at Technical University of Denmark.

The aim of the course is to introduce the students to some of the modern methods and algorithms used in Scientific Machine Learning (SciML), and let the students experience these methods on elementary computer experiments.

The PhD course covers several topics in SciML: neural differential equations, universal differential equations, physics-informed neural networks (PINN), automatic differentiation (AD) / differentiable programming, neural operators, symbolic regression, and more. The objective is to give the student an overview of the "tools" available and how they can be modified for particular SciML applications. The course is partly based on the lecture notes from MIT's 18.337 Parallel Computing and Scientific Machine Learning.

#### Learning objectives:

A student who has met the objectives of the course will be able to:

- Understand problems and questions addressed by SciML methods.
- Understand how methods are used as building blocks to address SciML questions.
- Be able to choose a suitable method depending on the situation and problem.
- Implement some of these methods in Julia.
- Skilfully perform numerical experiments and interpret the results.
- Setup and train neural differential equations and physics-informed neural networks.
- Identify and exploit the properties and structure of scientific knowledge when machine learning

DTU

MIT  
Massachusetts Institute of Technology

DCAMM

DTU Compute  
Institute for Mathematics and Computer Science

DDSA  
DANISH DATA SCIENCE ACADEMY

EuroTech Universities  
Institutes of Science and Technology

GPUlab  
DTU Compute

Lecturers

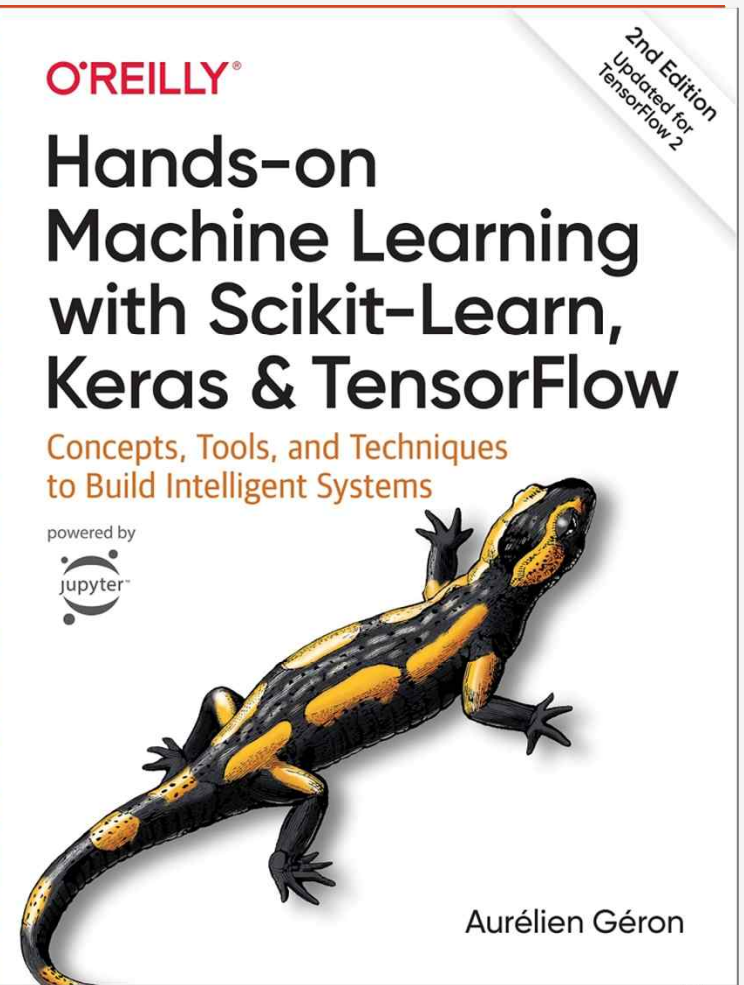
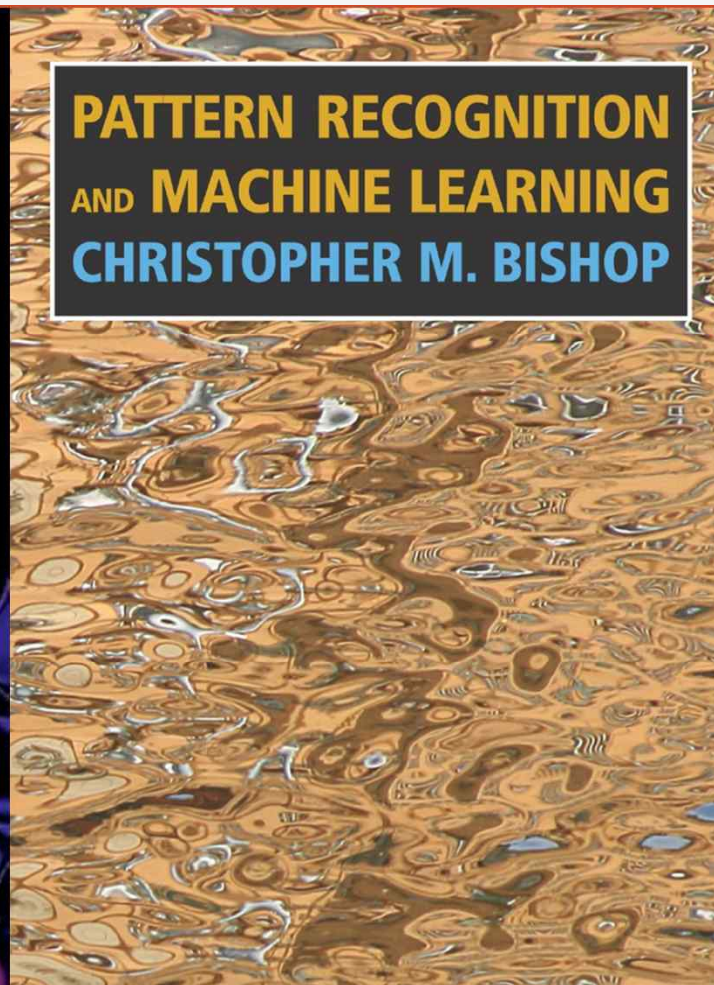
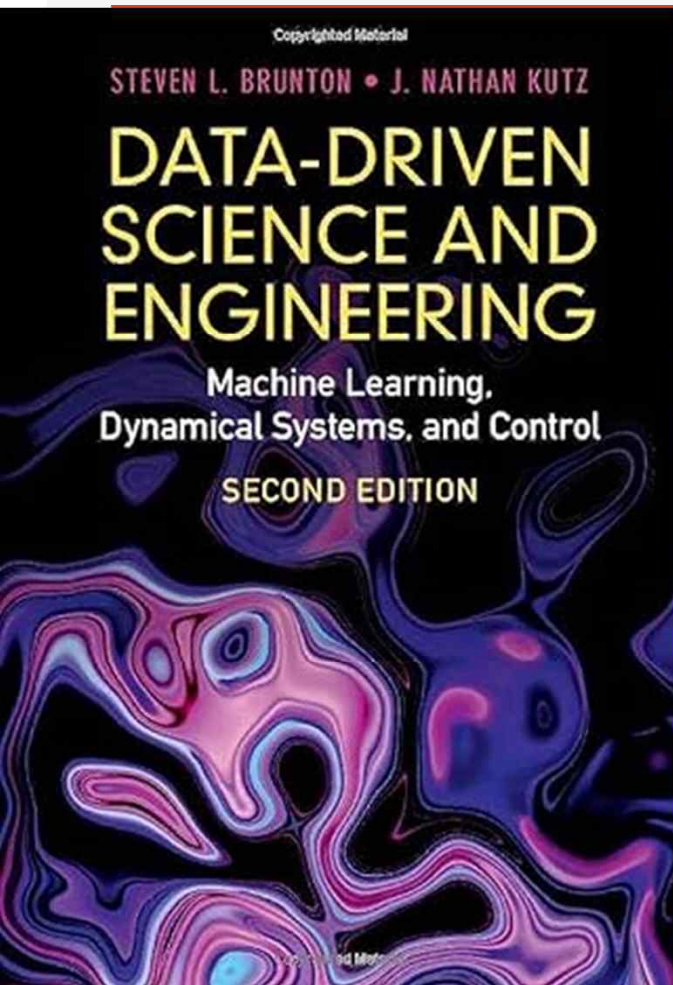
- Christopher Rackauckas

Organizer

- Allan P. Engsig-Karup

Teaching assistants

# References



---

Thank you



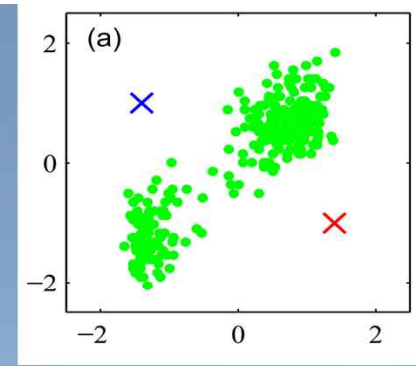
SGH

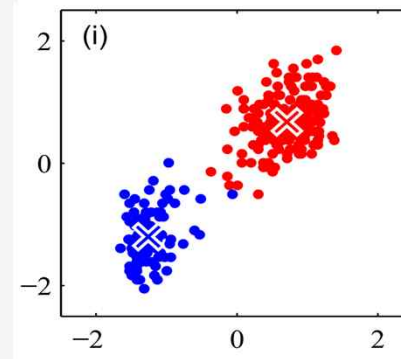
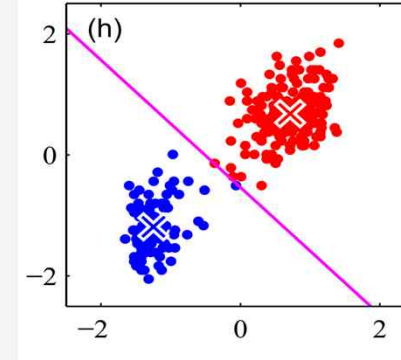
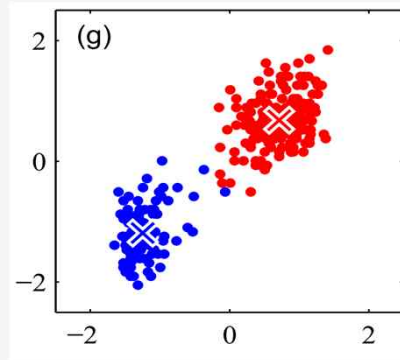
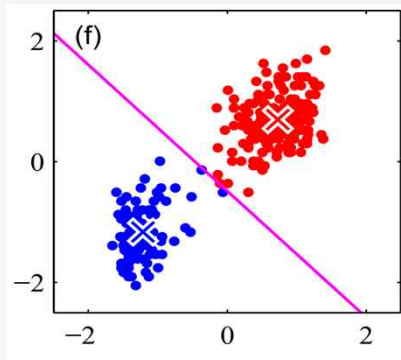
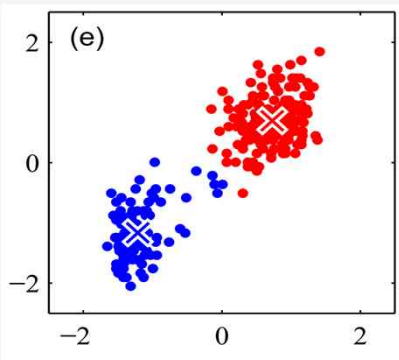
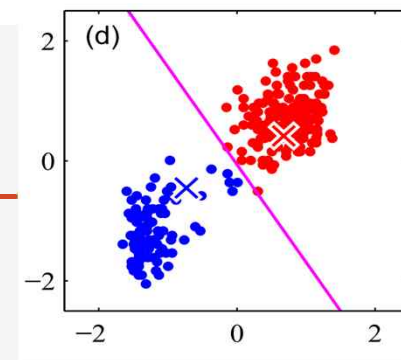
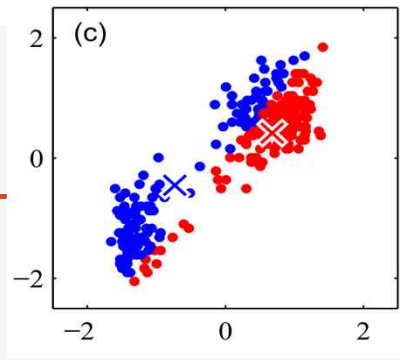
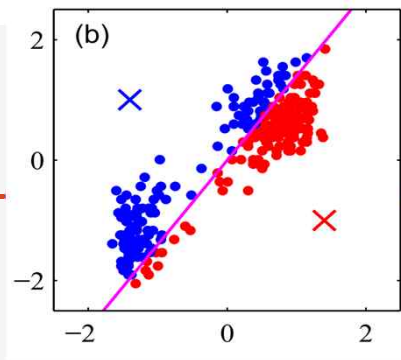
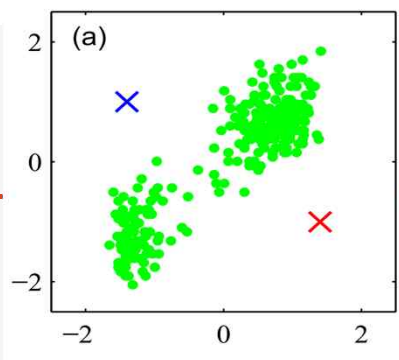


# Prerequisite

---

- Linear Algebra
- Probability
- Data Science
- Information Theory
- Programing
- Coding (Python.....)
- **Problem Setting**



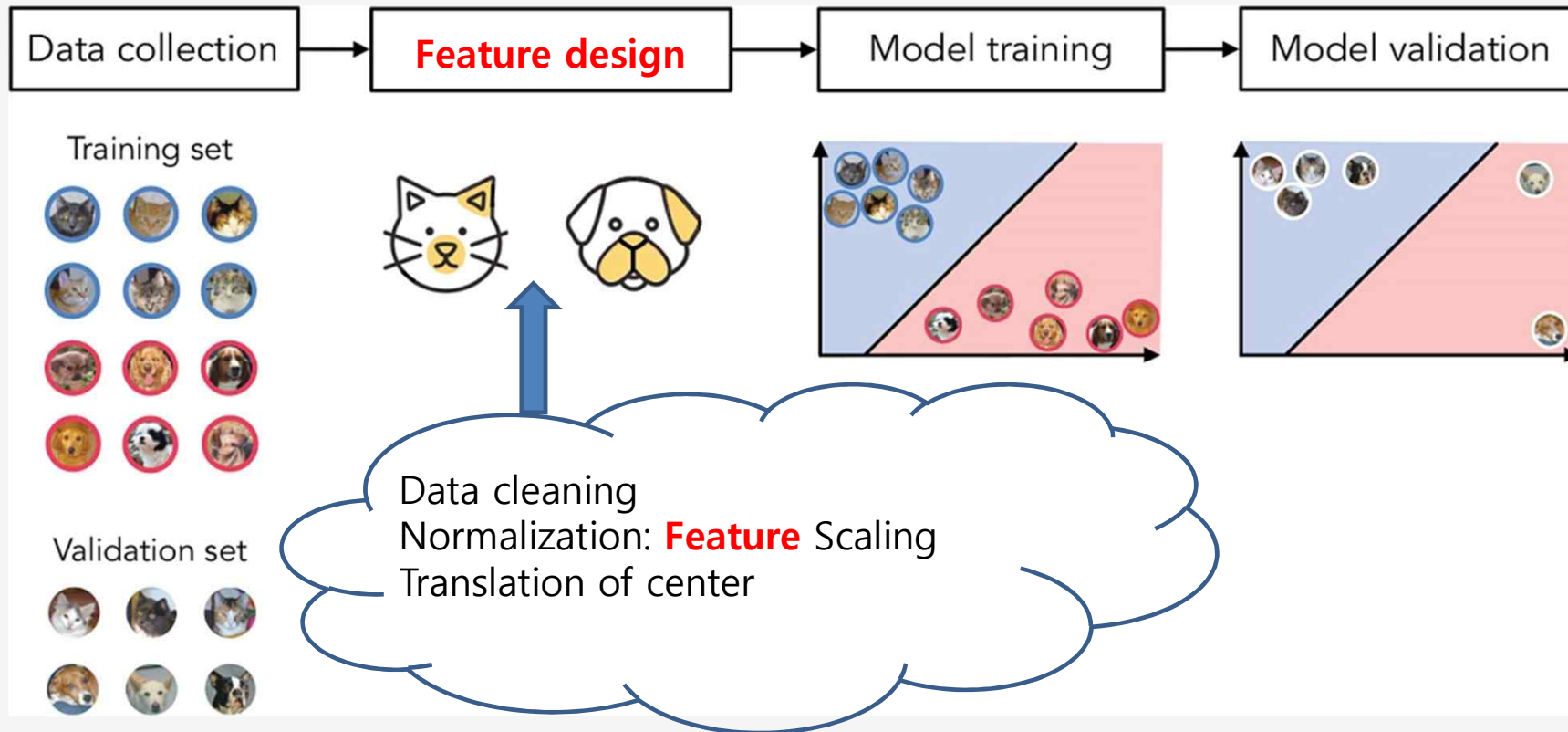


# Youtube 학습



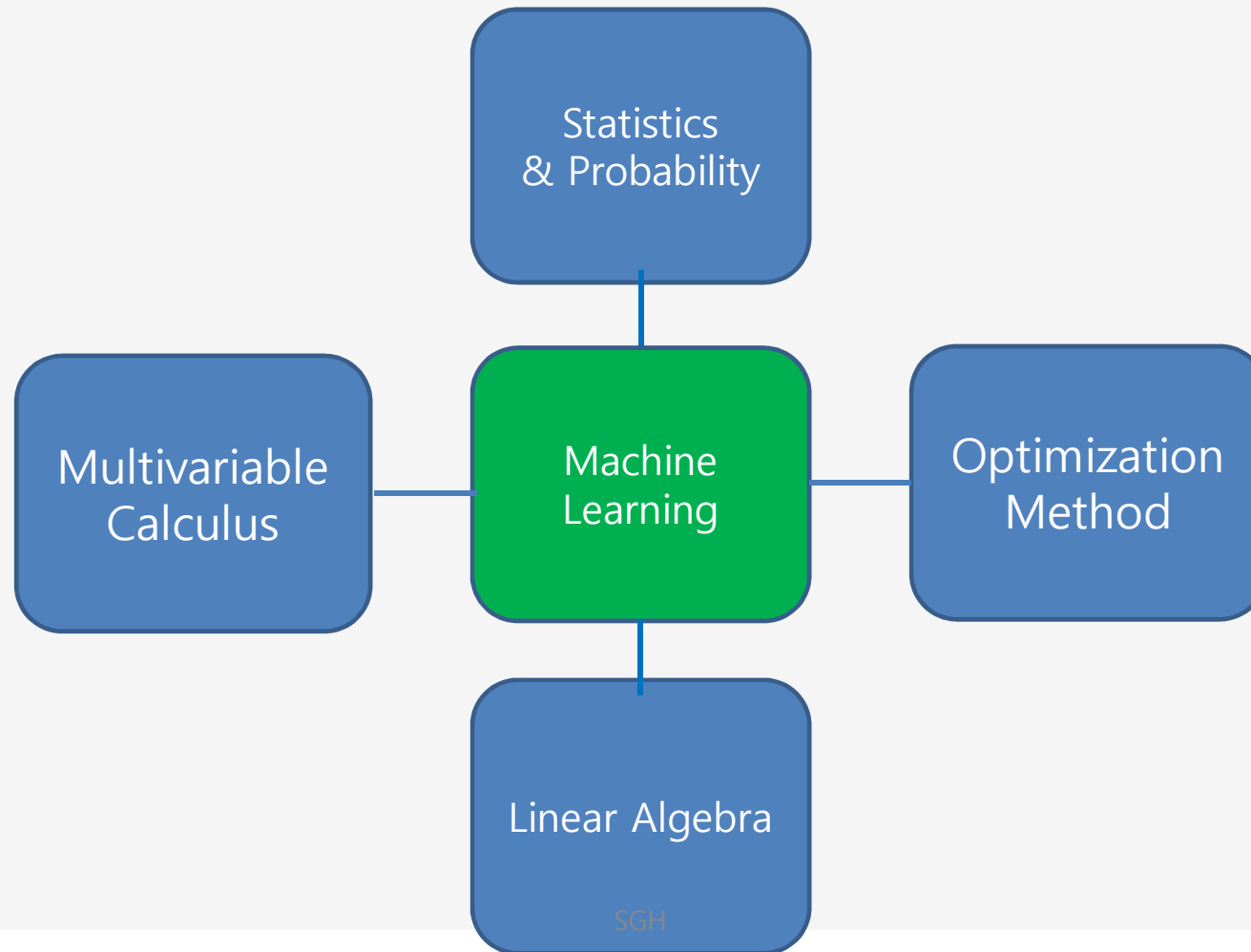
Subject	1	2	3	4
probability				
Linear algebra				
Data prep.				
ML algorithm				

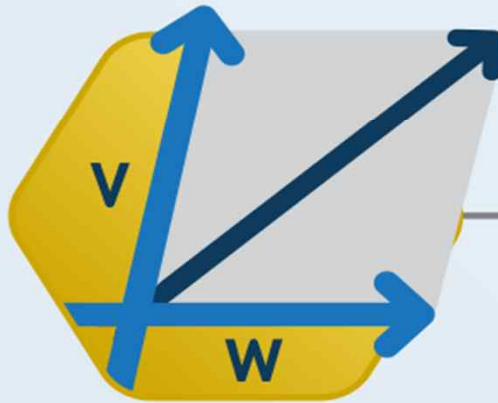
# ML Process



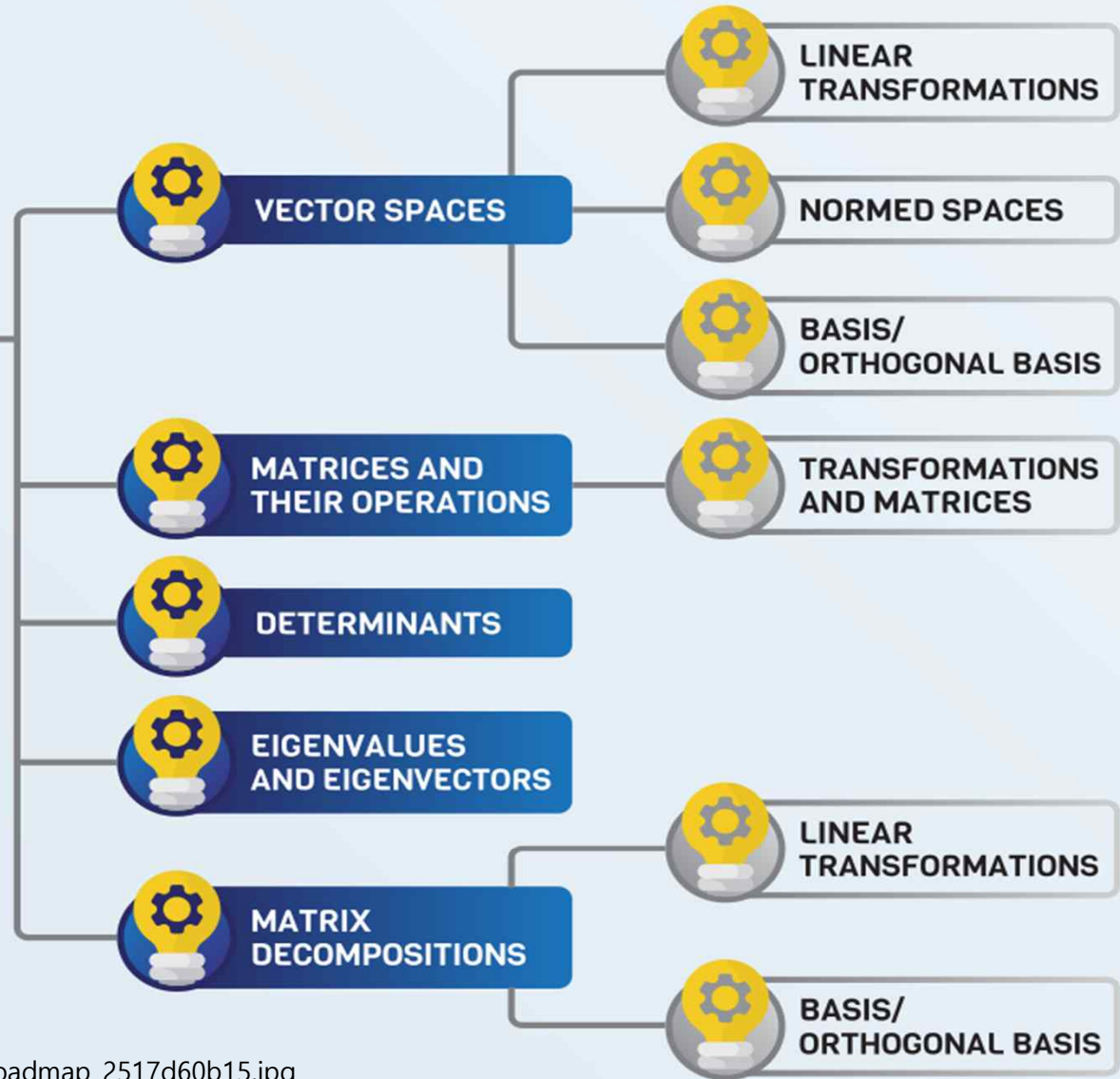
# Math Skills for Machine Learning

---

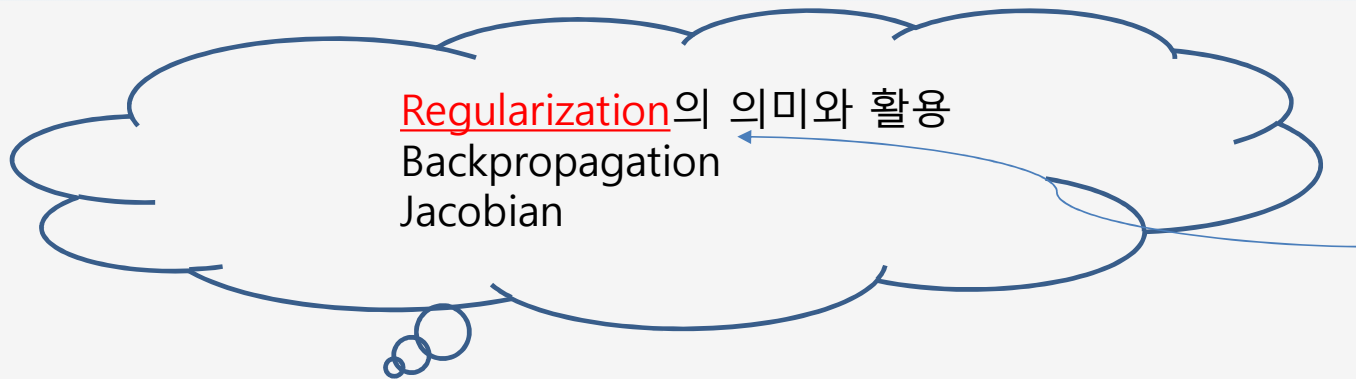
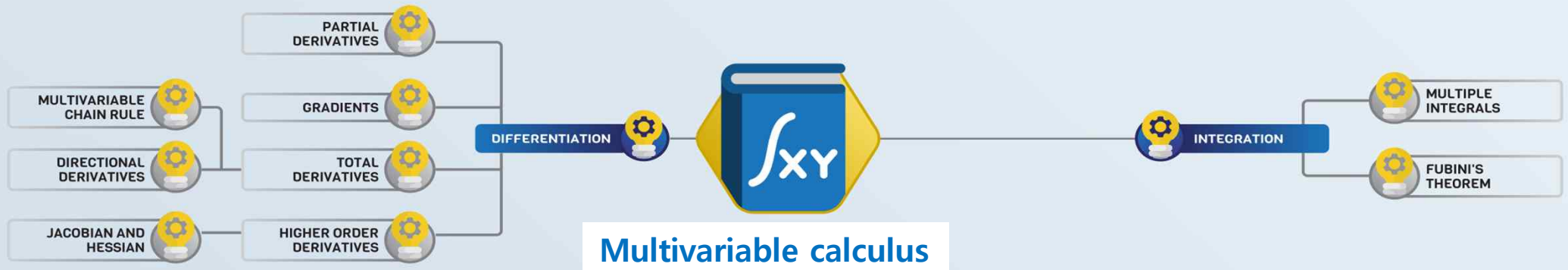




**Linear Algebra**



# Composite function의 이해



Magic of ML and DL



# NEURAL NETWORKS

