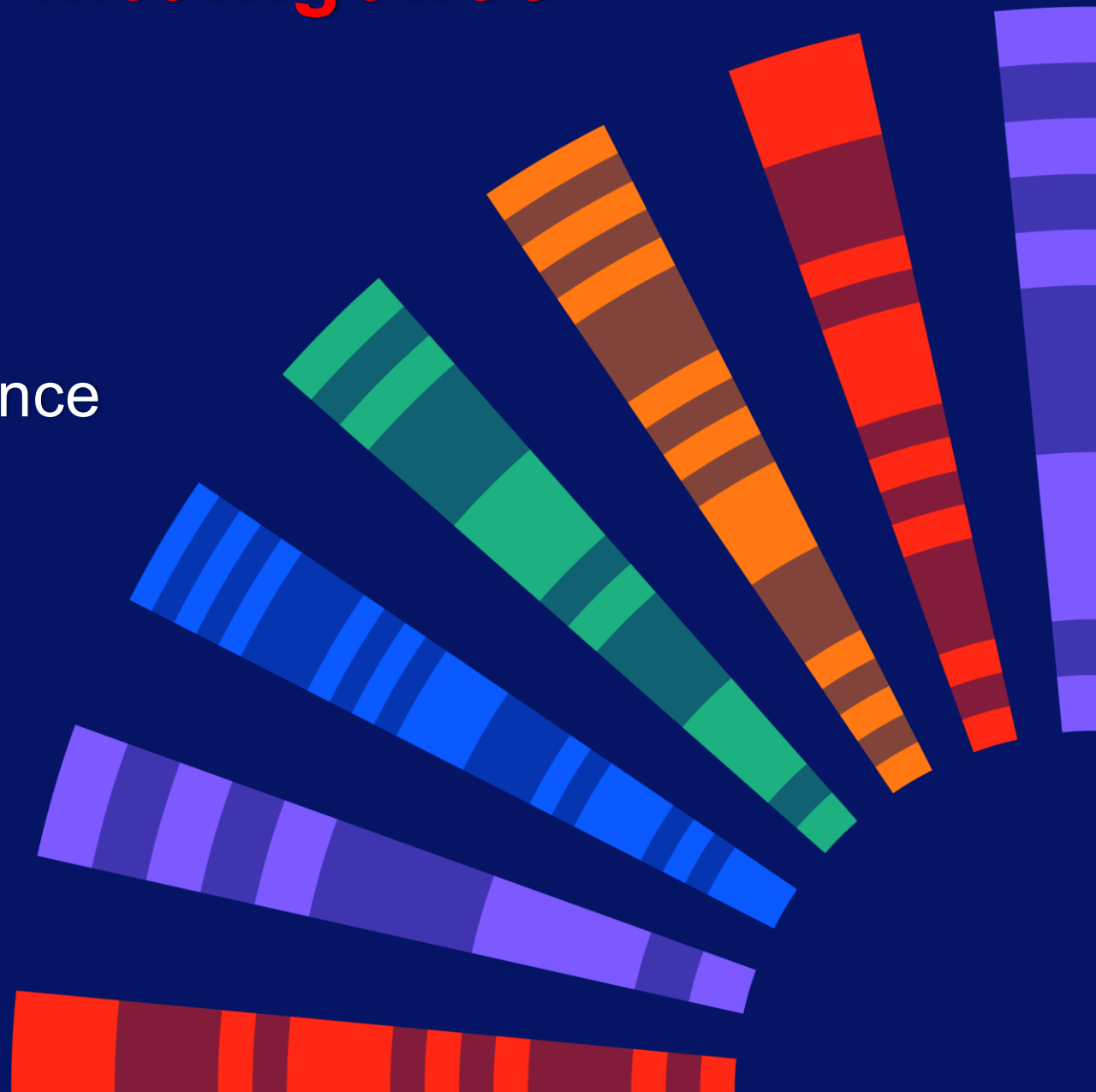
Pursuing the Nature of Intelligence

Professor Yi Ma

School of Computing and Data Science

The University of Hong Kong



SCHOOL OF COMPUTING & DATA SCIENCE
The University of Hong Kong

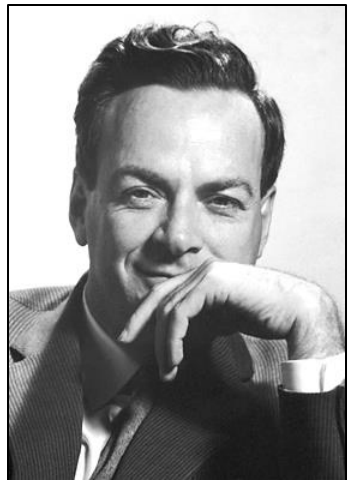
Prologue

Seek a scientific and theoretical foundation for Intelligence:

- **what to learn?**
- **how to learn?**
- **why correct?**

“What I cannot create, I do not understand.”

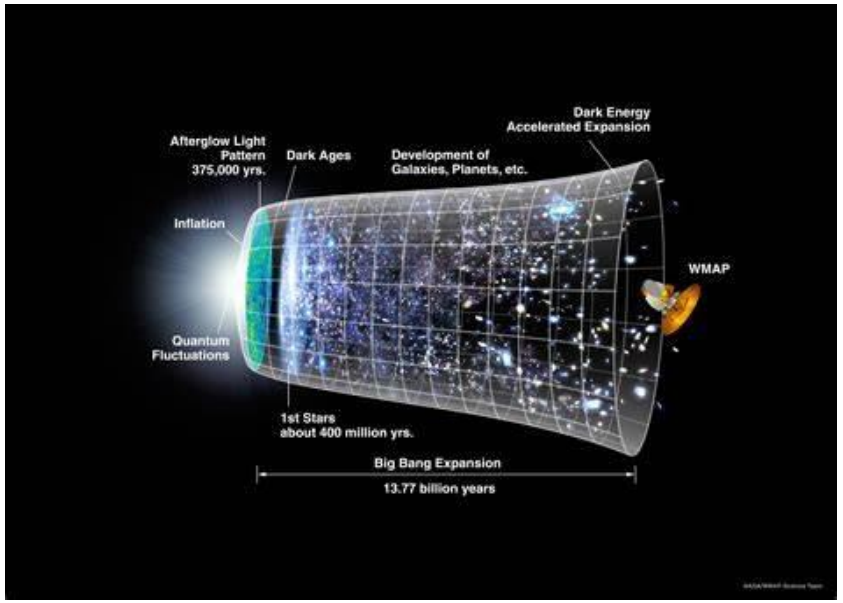
-- Richard Feynman



知致物格

Evolution of Life and Intelligence

Evolution of the Universe is Physics at work



Evolution of Life is Intelligence at work

"Just as the constant increase of entropy is the basic law of the universe, so it is the basic law of life to be ever more highly structured and to struggle against entropy."

-- Vaclav Havel

Evolution of Life and Intelligence: From DNA to Brain

From the first DNA to the emergence of life with Brain: **3.6 Billion Years**
From the first Brain to the explosion of lives in the Cambrian period: **50 Million Years**

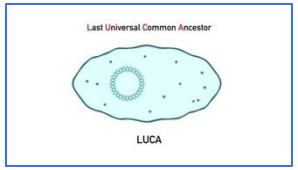
DNAs: Pretrained “Large Models”
Random Mutation & Natural Selection
Reinforcement Learning

DNA appeared near some volcano in the ocean

40B Years

First DNA

Common ancestor of all lives: **LUCA**”



3.5B Years

First Life Form

Emergence of Brain & Senses
Individual Models: Memory
Learn from Feedback

First life form with a brain (**Nematode**)

550M Years

First Brain

Explosion of Lives

The Cambrian period

500M Years

Cambrian Period

A Brief History of Intelligence, Max Bennett, 2023

Evolution of Life and Intelligence: From Species to Individuals

Emergence and evolution of life are mechanisms of intelligence at work!

Life depends on intelligence to continuously acquire more knowledge to better predict the world.

Phylogenetic Intelligence: DNA inheritance, random mutation, and natural selection

Ontogenetic Intelligence: Individual memory, perception & feedback, and error correction.








3.7B years ago
Life begins

500M years ago
Cambrian period

400M years ago
fish

360M years ago
amphibian

250M years ago
reptile

200M years ago
bird and mammal

310T years ago
neanderthal

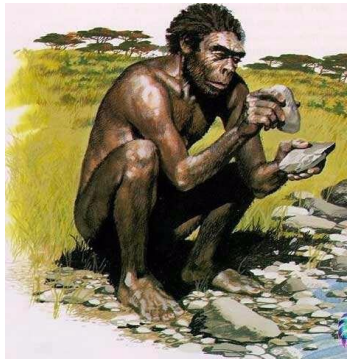
Evolution of Intelligence: From Societal to Artificial Intelligence

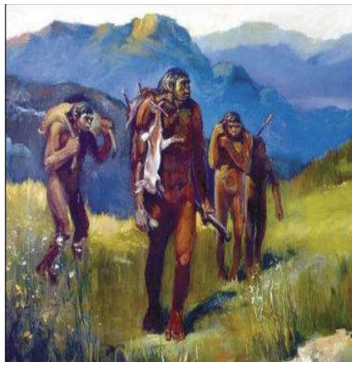
Emergence and evolution of life are mechanisms of intelligence at work!

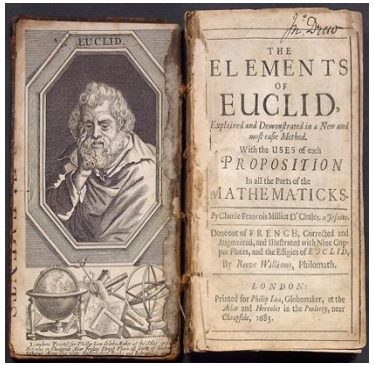
Life depends on intelligence to continuously acquire more knowledge to better predict the world.

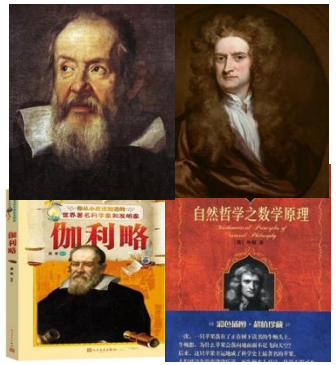
Societal Intelligence: Languages and texts, empirical knowledge, trial and error

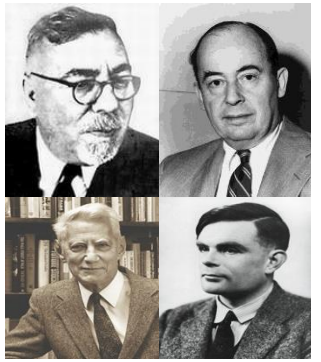
Artificial Intelligence: Scientific facts, theorize, hypothesis testing & falsification.









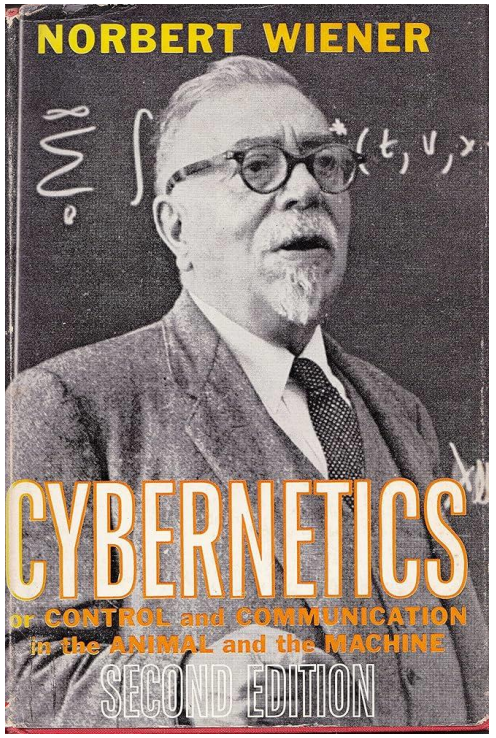


310T years ago	70T years ago	3500 BC	600-300 BC	14-18th Century	The 1940s
neanderthal	**Societal intelligence**	written language	**Artificial Intelligence**	Renaissance	**Machine Intelligence**
Tools and group hunting	Languages Information sharing	Knowledge accumulation	Abstraction, formal logic, and mathematics	Science Hypothesis Testing	Computing machines

The Origin of Machine Intelligence (the magic era!)

1940s, people started to make machines imitate intelligence (of animals).

- 1948, **Cybernetics & System Theory**, **Nobert Wiener**
- 1943, **Artificial Neural Networks**, Warren McCulloch and Walter Pitts
- 1948, **Information Theory**, Claude Shannon
- 1944, **Game Theory**, John von Neumann
- 1940's, **Turing Machine and Turing Test**, Alan Turing



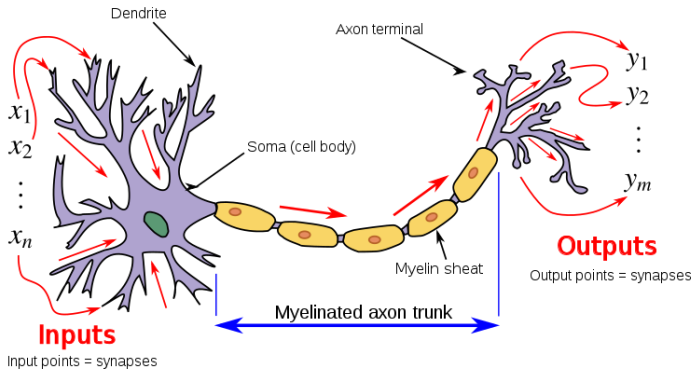
Perceive (encode)

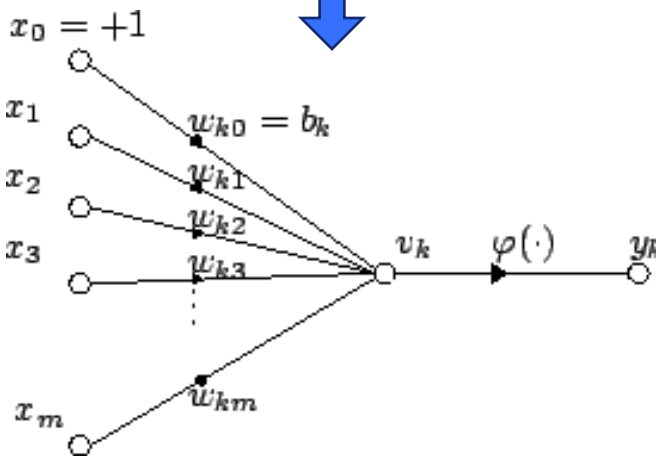
Predict (decode)

Learn memory

Artificial Neurons and Neural Networks: Learn from Nature

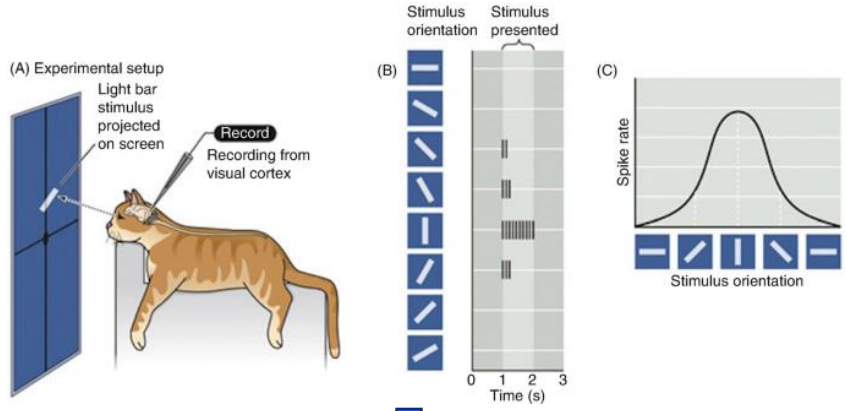
Golgi and Cajal 1888 (1901 Nobel Prize)

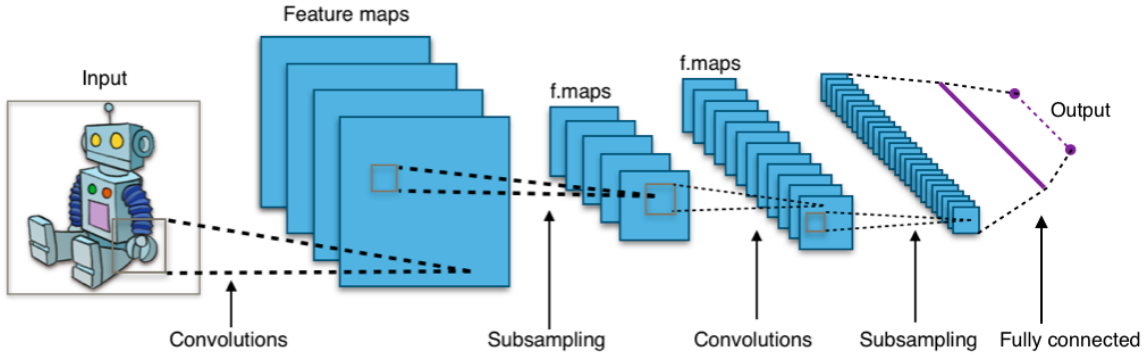




Warren McCulloch & Walter Pitts 1948

Hubel and Wiesel 1959 (1981 Nobel Prize)





Fukushima 1980 & LeCun 1989 (Turing Award)

History of Machine Intelligence (Artificial Neural Networks)

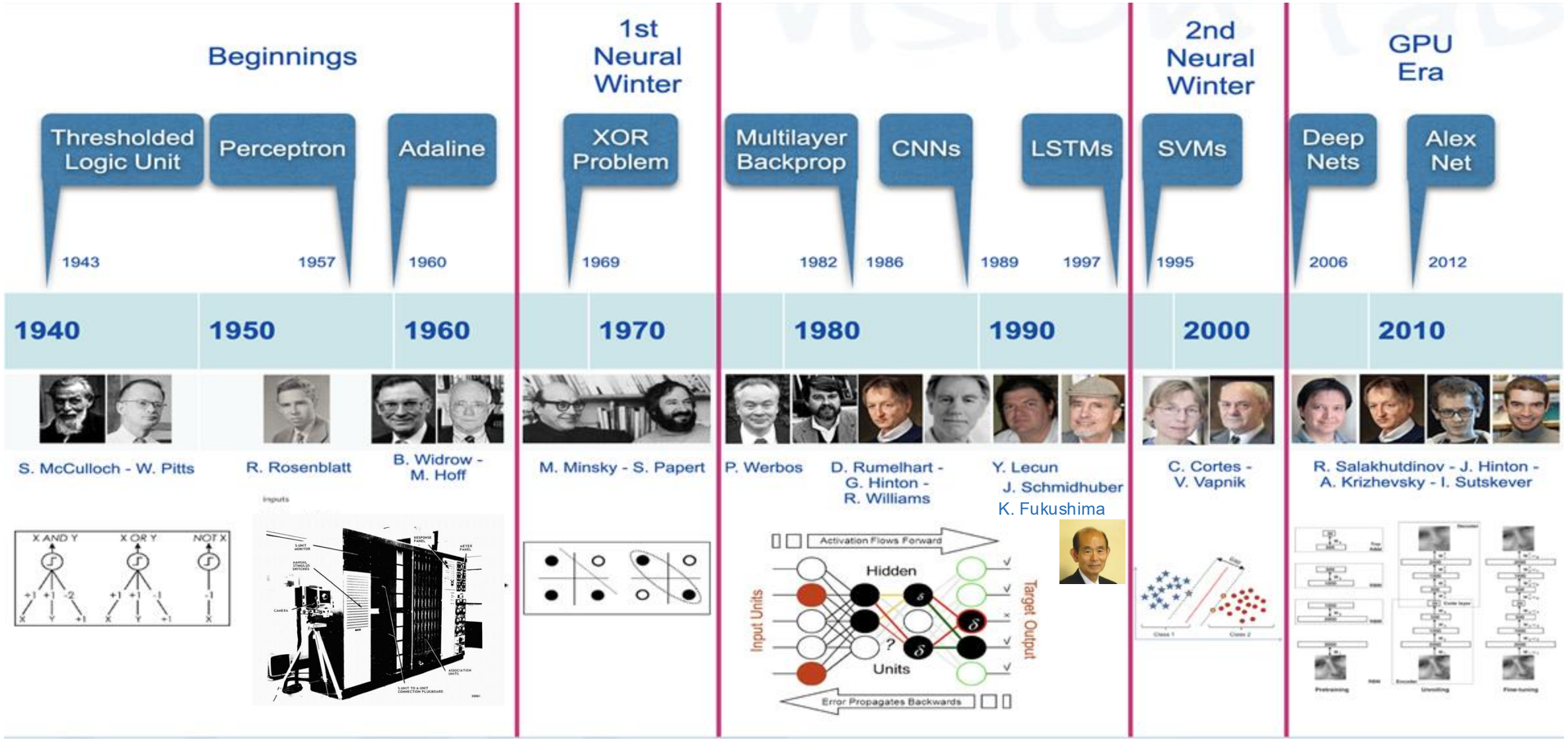
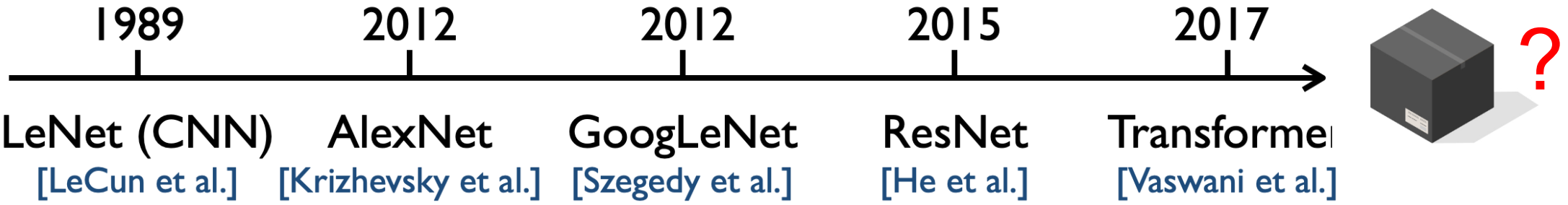
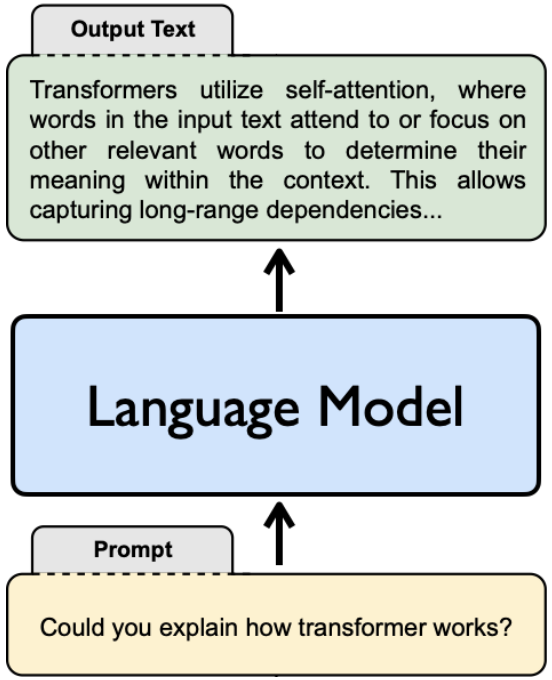


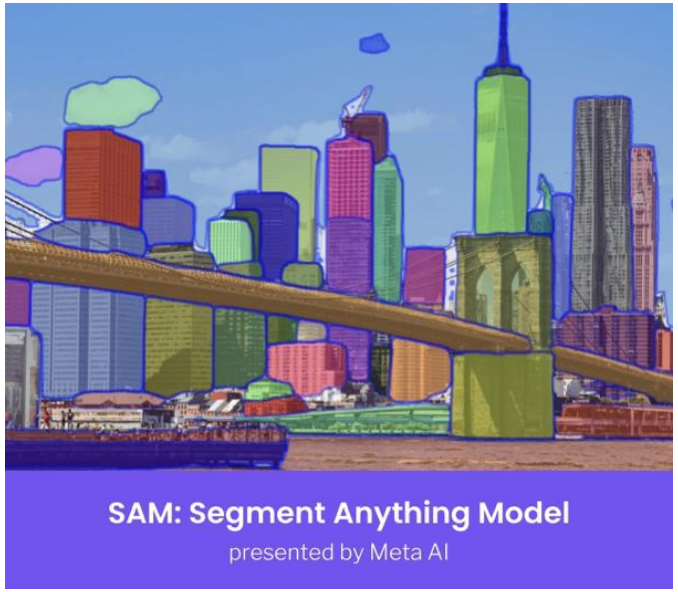
Figure courtesy of Professor Rene Vidal

Modern Evolution of Deep Neural Networks

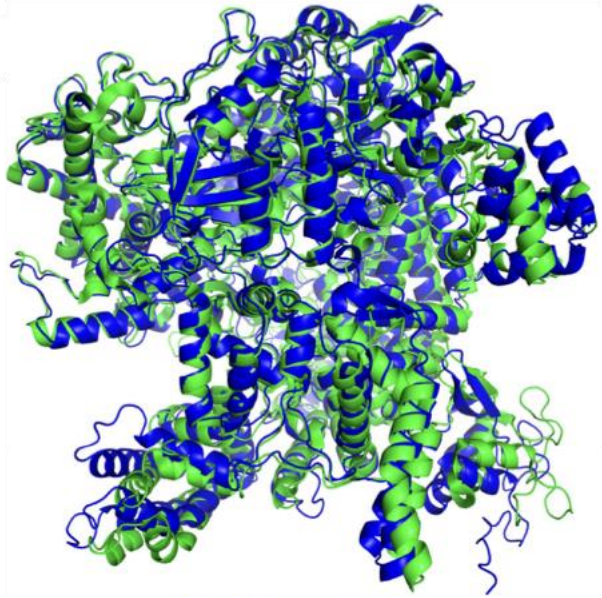
Deep Neural Nets







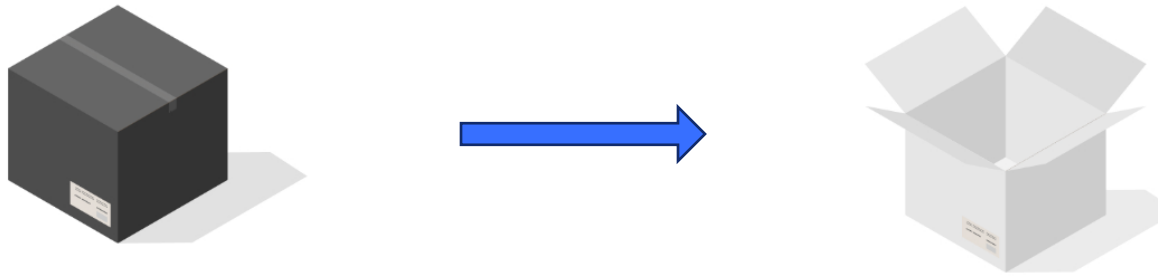
(2024 Nobel Prize Chemistry)



Why Must Turn Blackbox to Whitebox?

- Modern AI systems all based on empirically designed deep networks (alchemy?)
- Blackbox is difficult to explain, impossible to guarantee, costly to improve, …

It is high time to develop a principled approach!



What to Learn?
(Parsimony)

- Learn what is predictable
- Low-dim structures
- Information gain

How to Learn?
(Compression)

- Unroll iterative optimization
- Iterative compressing
- Make DNN a whitebox

Why Correct?
(Consistency)

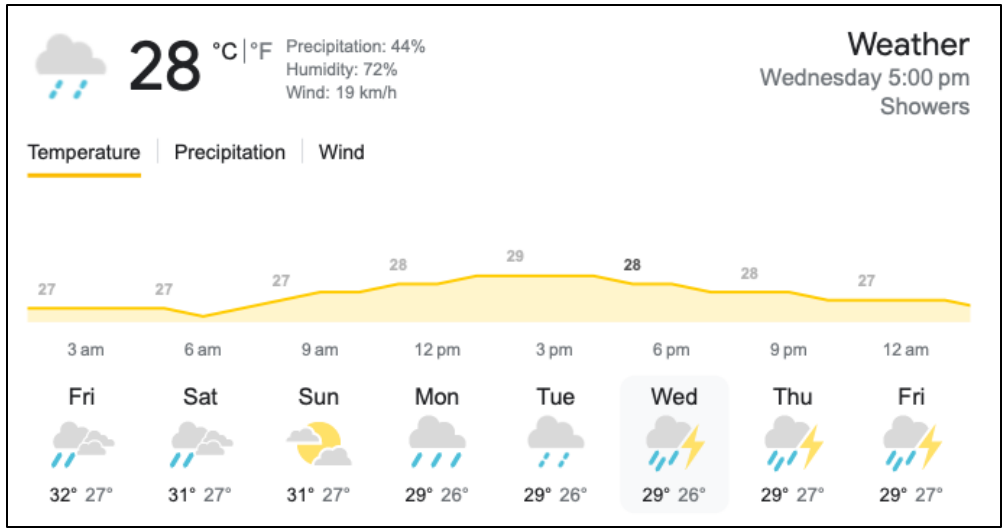
- Encoding & decoding
- Continuous learning
- Closed-loop feedback

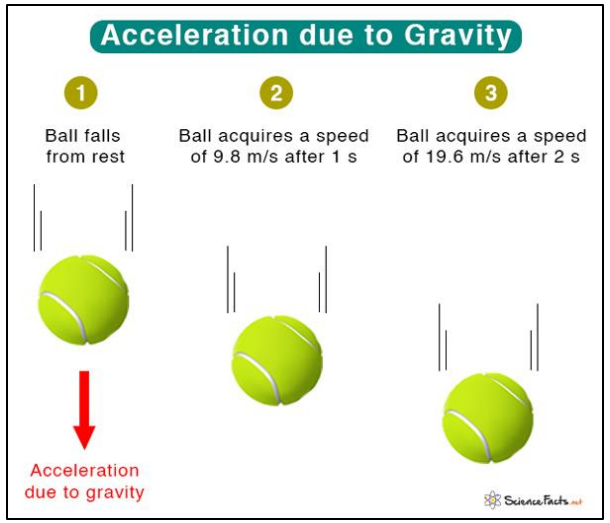
What to Learn?

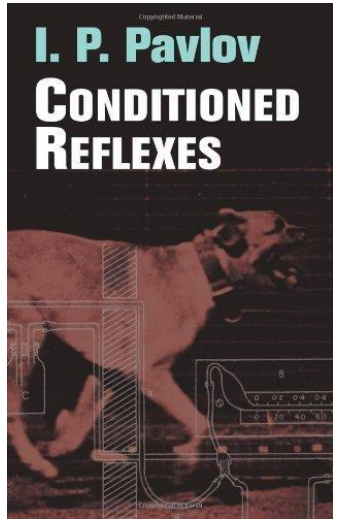
The fundamental reason why intelligence exists and evolves:

The world is not entirely random yet, and it is still largely predictable.

Intelligence and Science learn what is predictable from sensed data of external world (so every animal is Newton and has learned an accurate “world model.”)





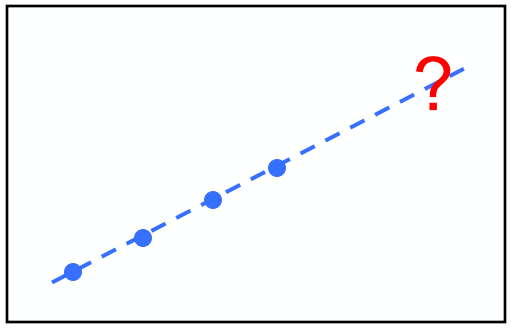


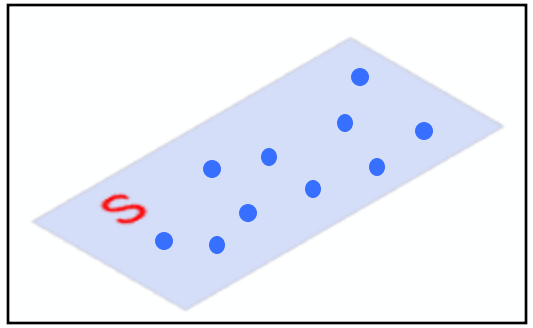
What to Learn?

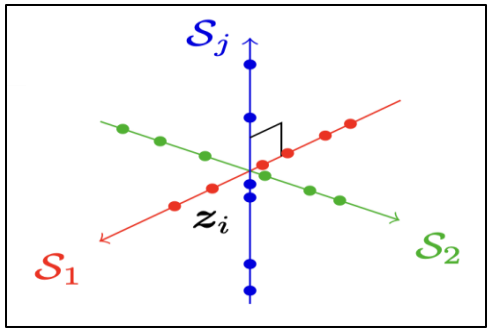
The fundamental reason why intelligence exists and evolves:

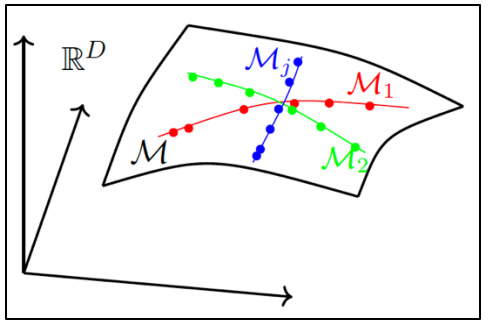
The world is not entirely random yet, and it is still largely predictable.

Mathematically, all predictable information is encoded as a distribution $p(\boldsymbol{x})$ of **low-dimensional** supports in observed **high-dimensional** data space.









This is the only "inductive bias" necessary!

High-dimensional Data Analysis with Low-Dimensional Models, Wright and **Ma**, 2022

What to Learn: Low-dimensionality

Important properties of low-dimensional structures: Completion

Bayes inference: $\boldsymbol{y} = \mathcal{P}_{\Omega}(\boldsymbol{x}) \rightarrow \underbrace{\hat{\boldsymbol{x}} \sim p(\boldsymbol{x} \mid \boldsymbol{y})}$

Conditional sampling

Partial observations

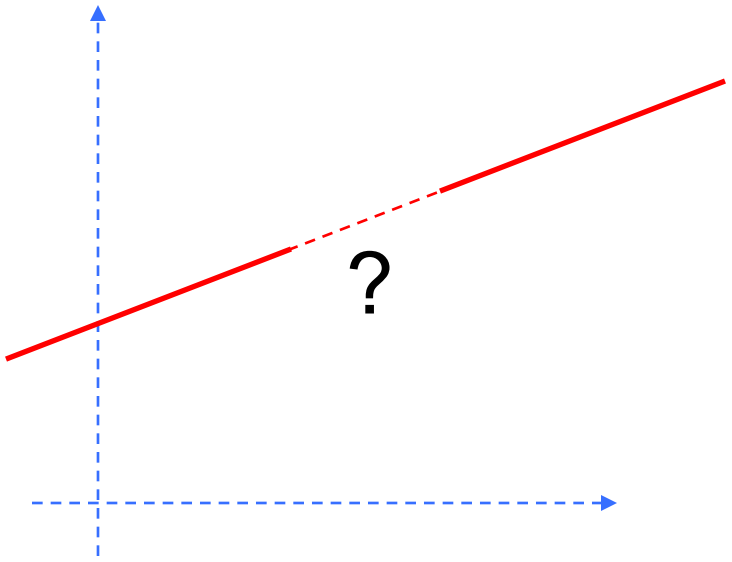
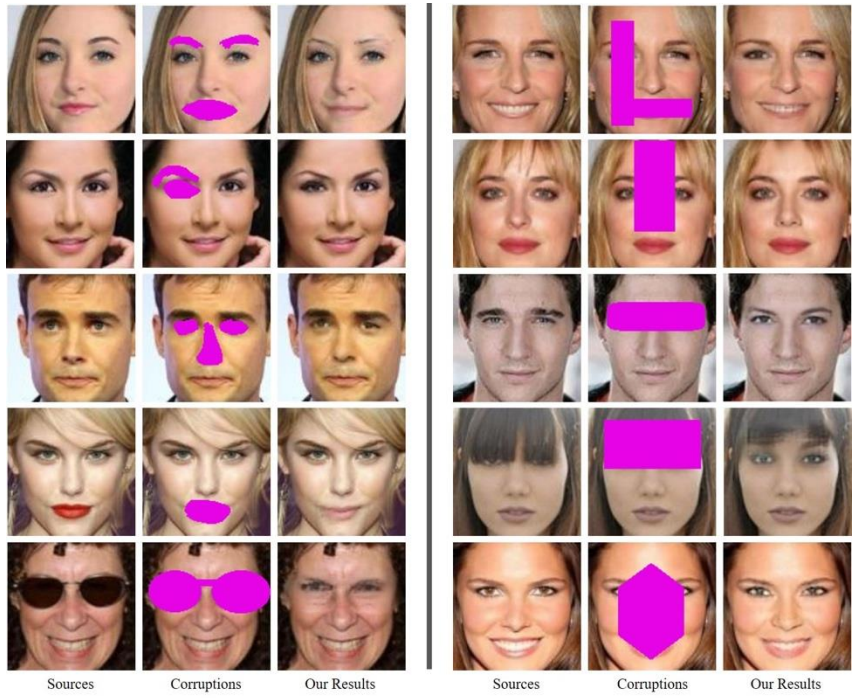
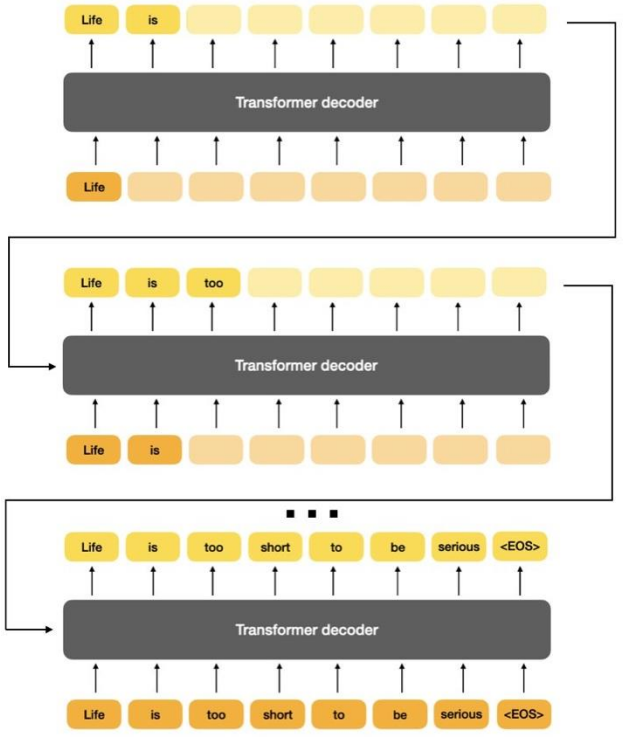


Image completion



Text prediction (GPT)



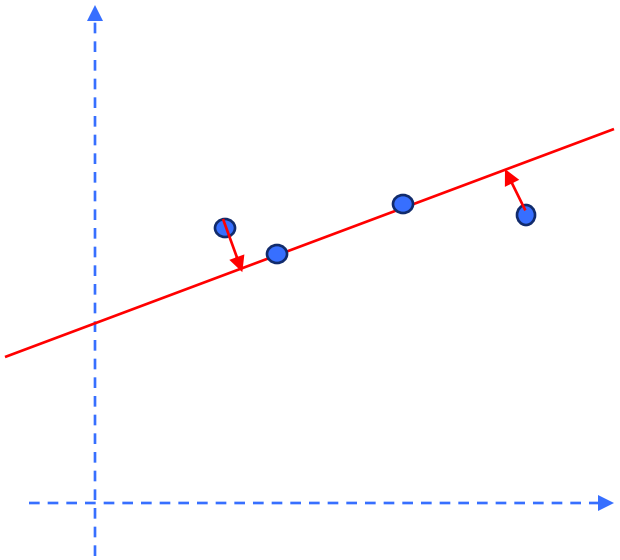
What to Learn: Low-dimensionality

Important properties of low-dimensional structures: Denoising

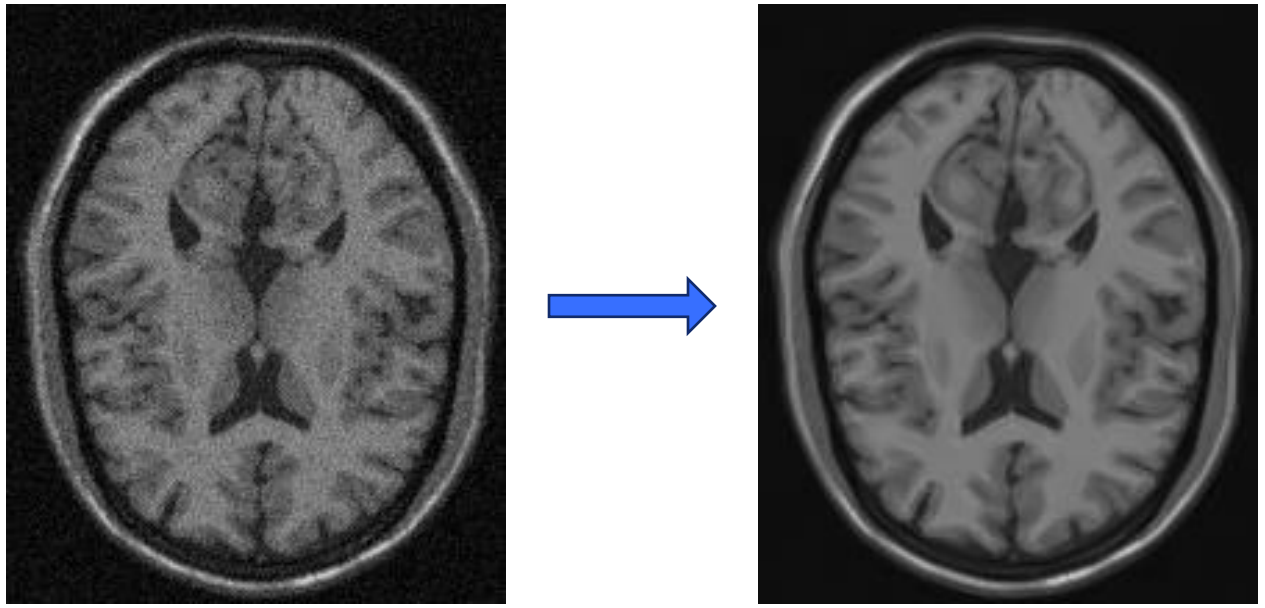
Empirical Bayes: $\boldsymbol{y} = \boldsymbol{x} + \sigma \boldsymbol{n} \implies \hat{\boldsymbol{x}} \sim \mathbb{E}(\boldsymbol{x} \mid \boldsymbol{y}) = \underbrace{\boldsymbol{y} + \sigma^2 \nabla \log p(\boldsymbol{x})}$

Tweedie's formula

Noisy observations



Natural image denoising (or generation)



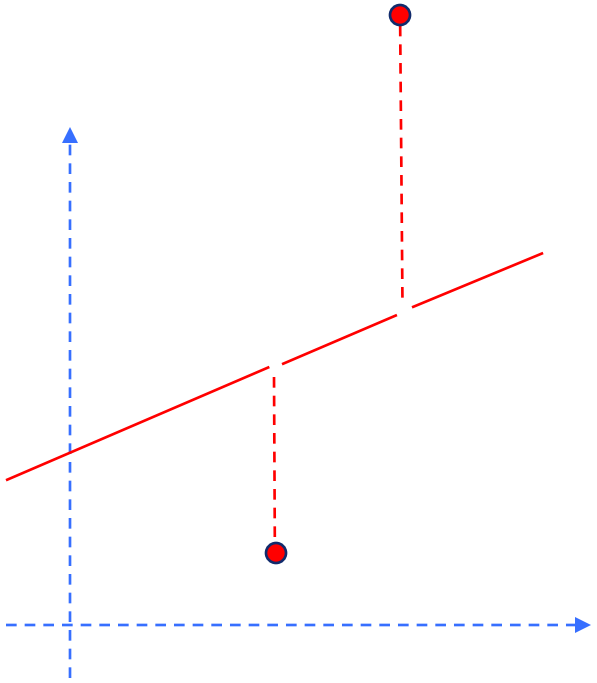
What to Learn?

Important properties of low-dimensional structures: Error Correction

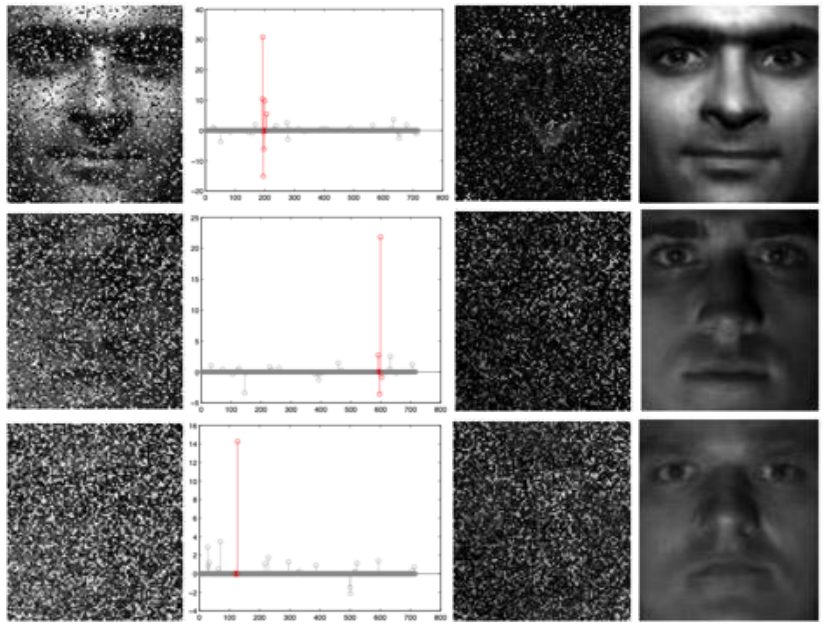
Robust Bayes inference: $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{e} \;\rightarrow\; \hat{\boldsymbol{x}} \sim \operatorname{argmin} \underbrace{||\boldsymbol{x}||_1 + ||\boldsymbol{e}||_1}$

Exploiting sparsity prior

Corrupted observations



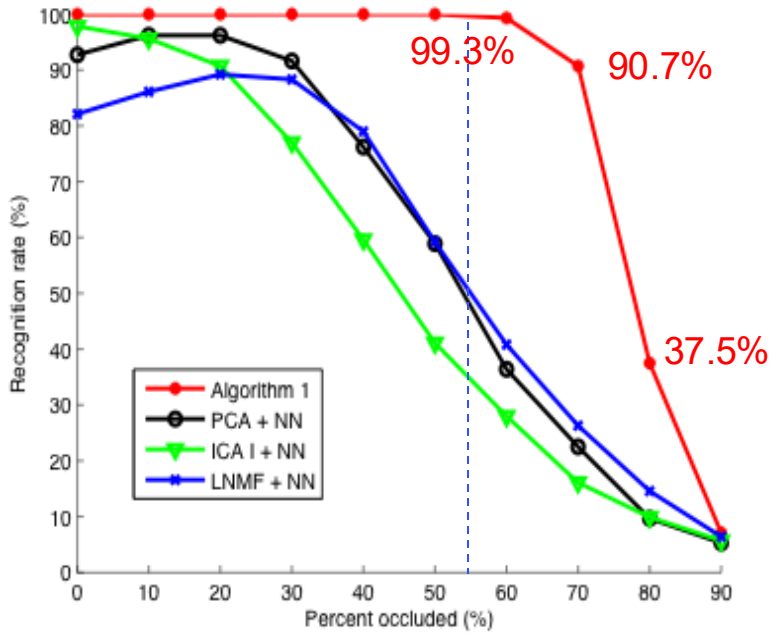
Robust (face) recognition via sparsity [WY+**Ma,** TPAMI'09]



30%

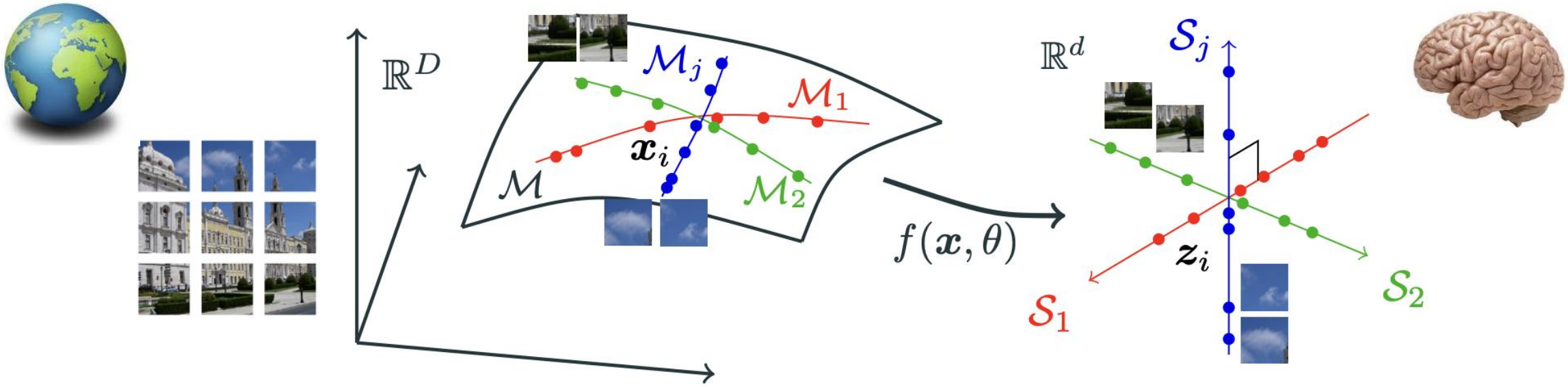
50%

70%



What to Learn: Seeking Parsimony

The main objective of learning:
Identify a distribution with low-dimensional structures from sensed data $\boldsymbol{x}$ and transform them to a compact and structured representation $\boldsymbol{z}$.

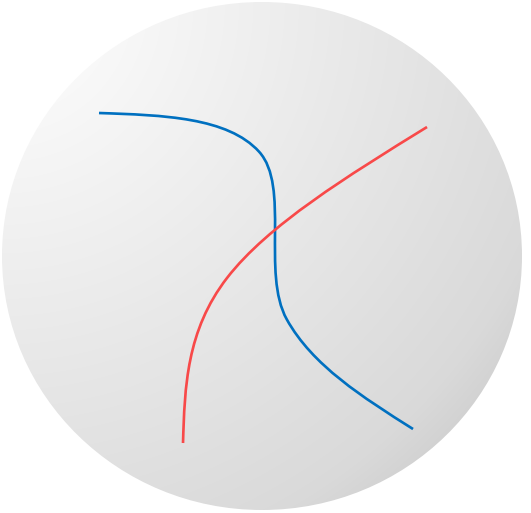


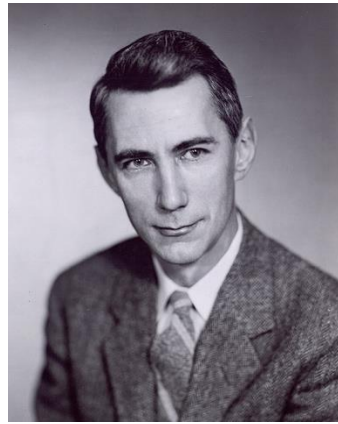
How to Learn: Measure of Parsimony

For distributions with **low-dim** structures, **entropy** (or "volume") of the underlying data distribution should be very small (or "zero"):

(Discrete) Entropy: $H(\boldsymbol{X}) = \sum_{\boldsymbol{x} \in \boldsymbol{X}} -p(\boldsymbol{x}) \log p(\boldsymbol{x})$

Differential Entropy: $h(\boldsymbol{x}) = \int -p(\boldsymbol{x}) \log p(\boldsymbol{x}) \, d\boldsymbol{x}$



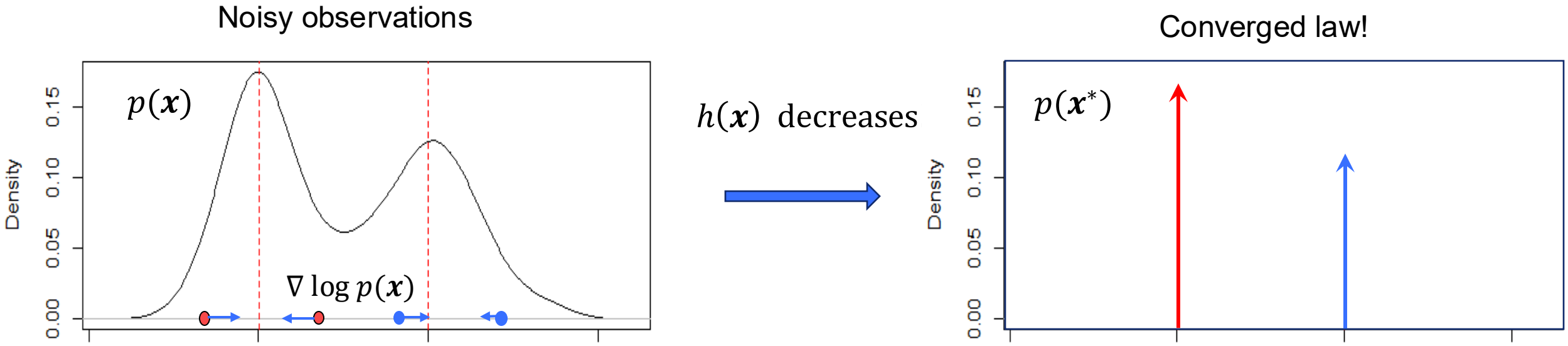


How to Learn: Compress

A fundamental and unifying mechanism to learn low-dim structures:

compress to reduce **entropy** of the observed (noisy) data distribution.

$$\text{minimize} \;\; h(\boldsymbol{x}) = \int -p(\boldsymbol{x}) \log p(\boldsymbol{x}) \, d\boldsymbol{x}$$



$$\boldsymbol{x}^{l+1} = \boldsymbol{x}^l + \beta \, \nabla \log p(\boldsymbol{x}^l)$$

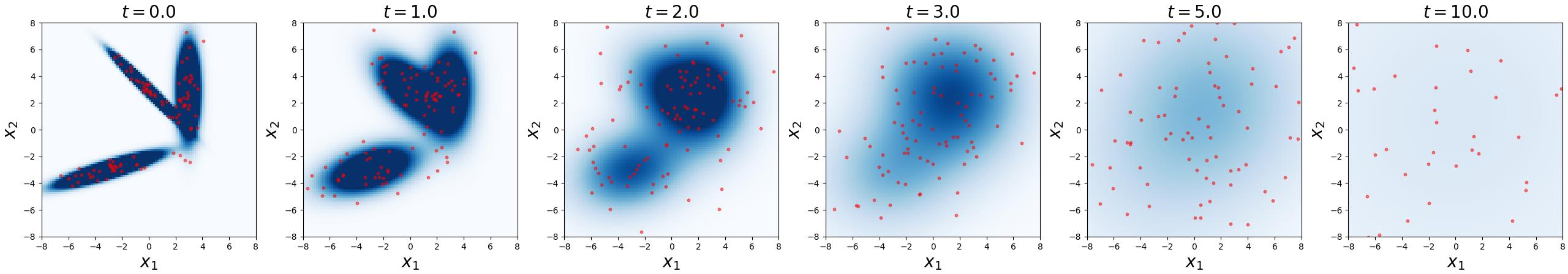
How to Learn: Compress via Denoising

Theorem **[Diffusion]** Consider the diffusion process:

$$\boldsymbol{x}_t = \boldsymbol{x}_o + t\boldsymbol{n}, \qquad \boldsymbol{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

Under natural technical assumptions, the **entropy of the process increases**:

$$\frac{d}{dt} h(\boldsymbol{x}_t) > 0, \qquad \forall\, t > 0.$$



Learning Deep Representations of Data Distributions, **Ma**+PBWY, 2025

How to Learn: Compress via Denoising

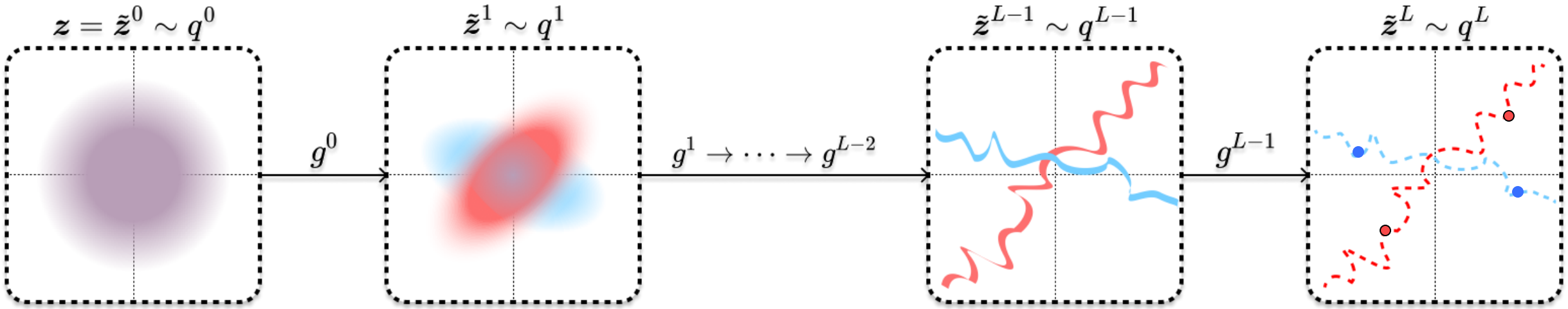
Theorem **[Denoising]** Consider the inverse denoising process:

$$\hat{\boldsymbol{x}}_{t-s} = \mathbb{E}[\,\boldsymbol{x}_{t-s} \mid \boldsymbol{x}_t\,] = \boldsymbol{x}_t + st\nabla \log p(\boldsymbol{x}_t)$$

Under natural technical assumptions, the **entropy of the process decreases**:

$$\frac{d}{ds} h(\hat{\boldsymbol{x}}_{t-s}) < 0, \qquad \forall\, s > 0.$$

$$\tilde{\boldsymbol{z}}^{l+1} = \tilde{\boldsymbol{z}}^{l} + \beta\, \nabla \log p(\tilde{\boldsymbol{z}}^{l})$$

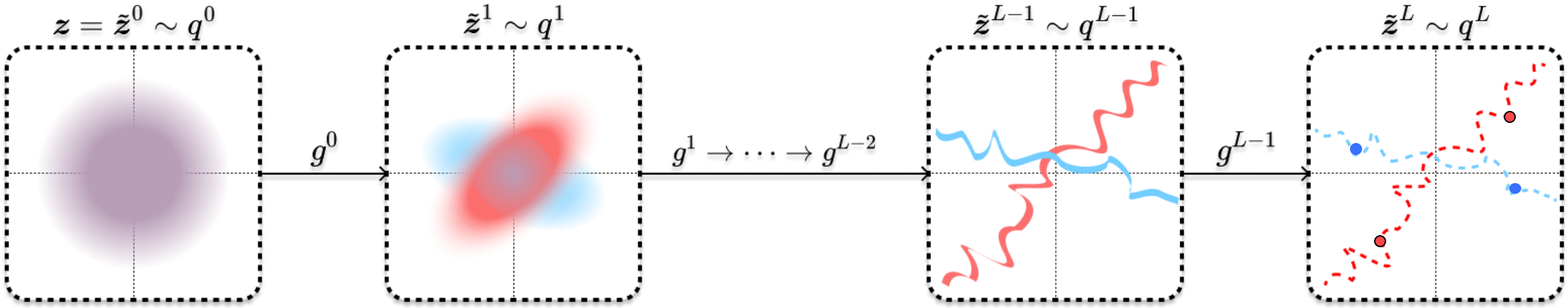


Learning Deep Representations of Data Distributions, **Ma**+PBWY, 2025

How to Learn: Empirical Approaches

Compression via iterative denoising

$$\tilde{\boldsymbol{z}}^{l+1} = \tilde{\boldsymbol{z}}^{l} + \beta \nabla \log p(\tilde{\boldsymbol{z}}^{l})$$

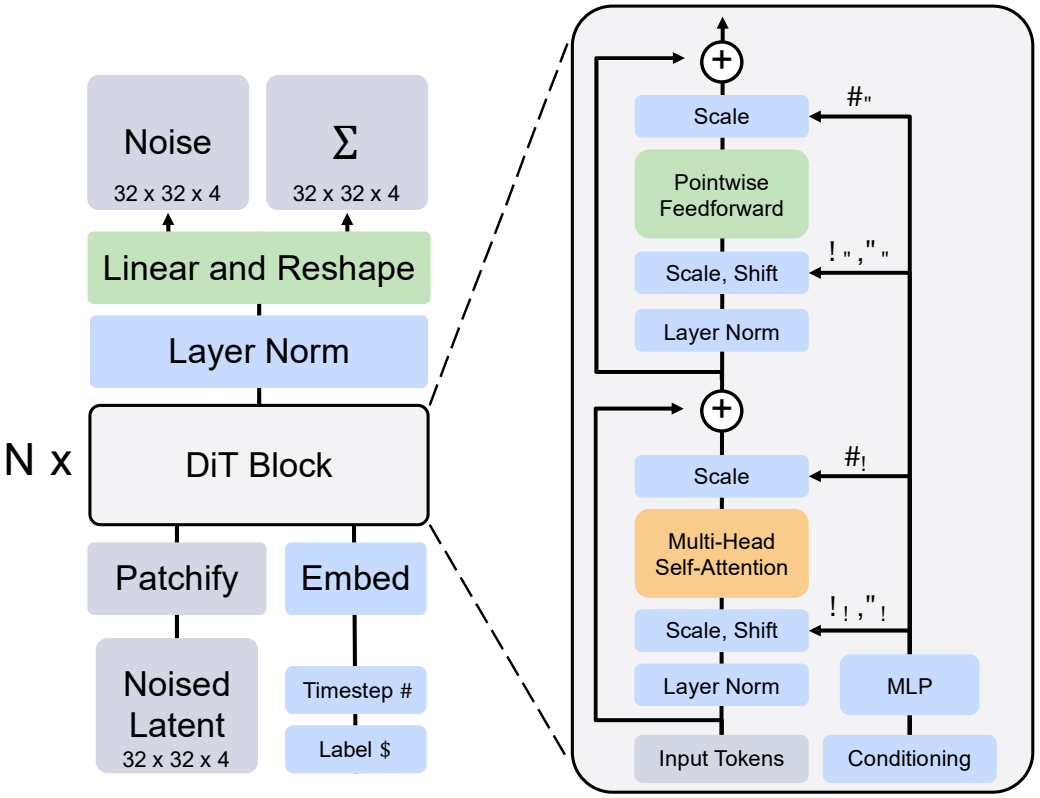


How to Learn: Empirical Approaches

Empirically designed networks to realize the denoising operator:

$$\tilde{\boldsymbol{z}}^{l+1} = \tilde{\boldsymbol{z}}^{l} + \beta \nabla \log p(\tilde{\boldsymbol{z}}^{l})$$ how to realize?

Diffusion Transformer (DiT)



Latent Diffusion Transformer

DiT Block with adaLN-Zero

Image or video generation

How to Learn: An Analytical Case

Analytically derived operation to realize the denoising operator:

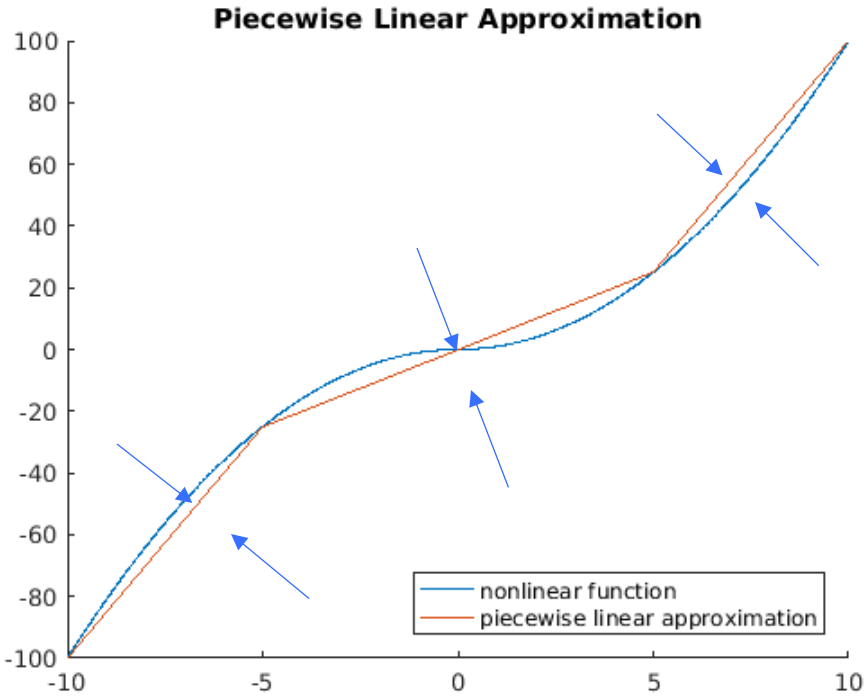
$$\tilde{\boldsymbol{z}}^{l+1} = \tilde{\boldsymbol{z}}^{l} + \beta \nabla \log p(\tilde{\boldsymbol{z}}^{l})$$ how to realize?

If we approximate a general distribution $p(\tilde{\boldsymbol{z}})$ with a **mixture of subspaces or low-dim Gaussians,** e.g. PCA, ICA, GPCA, Sparse Coding [W+**Ma,** 2022],

$$\tilde{\boldsymbol{z}}^{l} \sim \frac{1}{K}\sum_{k=1}^{K} \mathcal{N}(\boldsymbol{0}, \boldsymbol{U}_k \boldsymbol{U}_k^{\top}), \qquad \boldsymbol{U}_k \in O(D, d),$$

then

$$\tilde{\boldsymbol{z}}^{l+1} \propto \frac{1}{K}\sum_{k=1}^{K} \mathrm{softmax}\{\alpha ||\boldsymbol{U}_i^{\top}\tilde{\boldsymbol{z}}^{l}||^2\}_k \, \boldsymbol{U}_k \boldsymbol{U}_k^{\top}\tilde{\boldsymbol{z}}^{l}.$$



How to Learn: Measure of Information Gain

How to explicitly represent a distribution with a low-dimensional support?

A key idea: **lossy encoding and decoding**

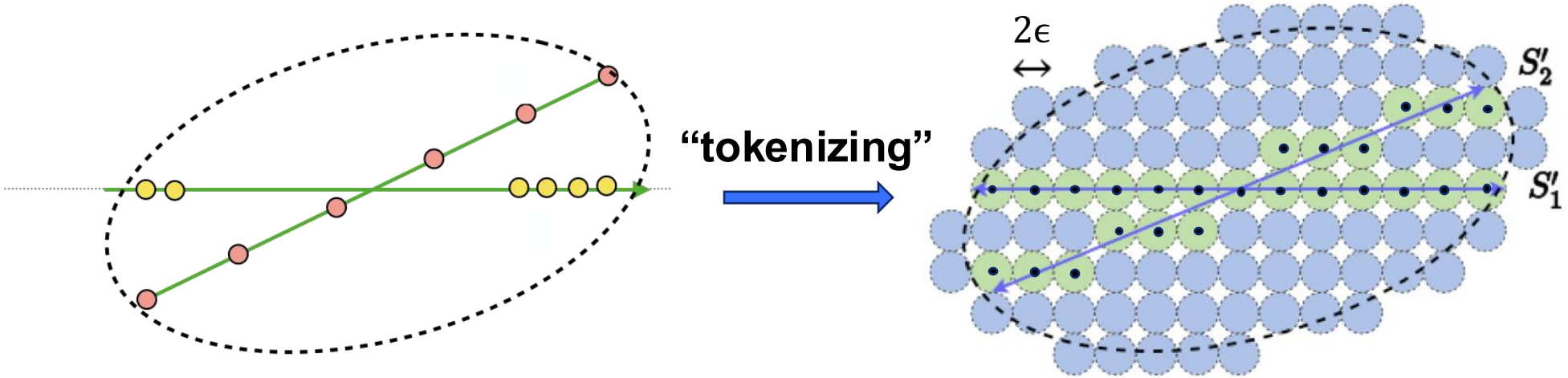
construct a finite codebook by packing the support of the distribution with ϵ-balls.

differential entropy

$$h(\boldsymbol{x}) = -\infty$$

rate distortion

$$R(\boldsymbol{x}, \epsilon) = \min_{\mathbb{E}||\hat{\boldsymbol{x}}-\boldsymbol{x}||\le\epsilon} h(\hat{\boldsymbol{x}}) - h(\hat{\boldsymbol{x}} \mid \boldsymbol{x})$$

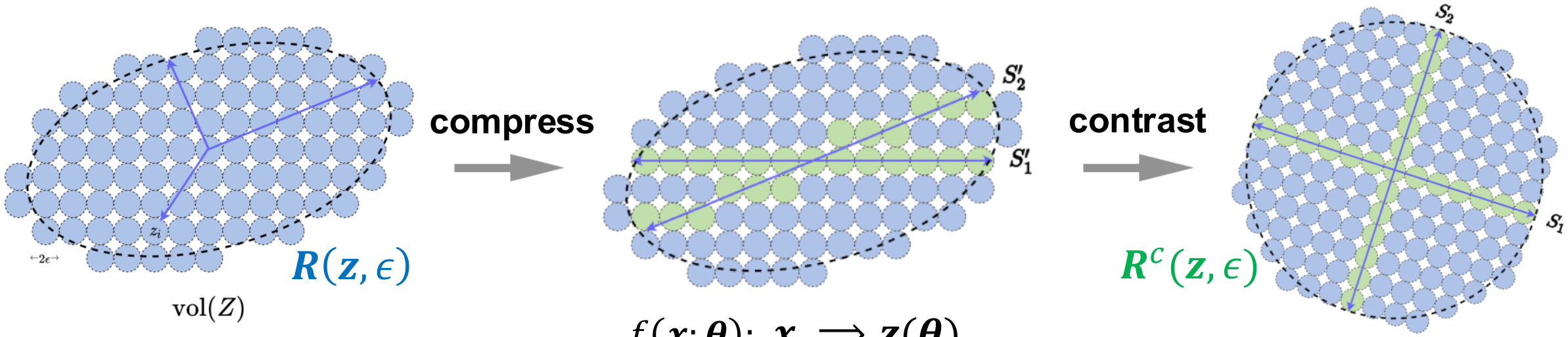


Segmentation of Mixed Data via Lossy Coding and Compression [**Ma**+DHW, TPAMI2007].

How to Learn: Organize Information

How to make the resulting representation the most informative?

compress what is similar; **contrast** what is dissimilar.



$$f(\boldsymbol{x};\boldsymbol{\theta}):\ \boldsymbol{x} \implies \boldsymbol{z}(\boldsymbol{\theta})$$

Mathematically: maximize information gain or reduce coding rate:

$$\max \Delta \boldsymbol{R}(\boldsymbol{z},\epsilon) = \boldsymbol{R}(\boldsymbol{z},\epsilon) - \boldsymbol{R}^{\boldsymbol{C}}(\boldsymbol{z},\epsilon)$$

ReduNet: A White-box Deep Network from the Principle of MCR^2 [YCY+**Ma**, JMLR2022].

How to Learn: Maximal Coding Rate Reduction (ReduNet)

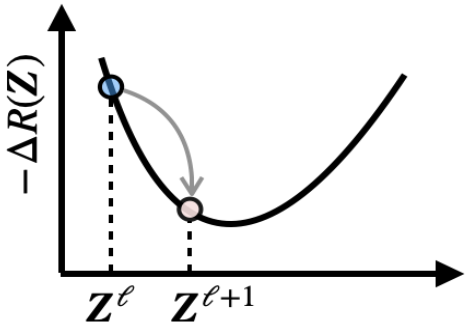
When $\boldsymbol{Z} = f(\boldsymbol{X}; \boldsymbol{\theta})$ is a mixture of K Gaussians with $\boldsymbol{\Pi}_k$ encodes membership of samples in the kth class.

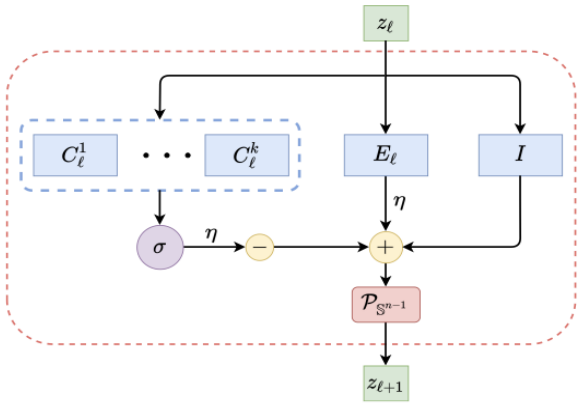
Maximal Coding Rate Reduction (MCR^2)

$$\max_f \Delta R(\boldsymbol{Z} \mid \boldsymbol{\Pi}) = \underbrace{\frac{1}{2}\log\det\left(\boldsymbol{I} + \frac{d}{\varepsilon^2}\cdot\frac{\boldsymbol{Z}\boldsymbol{Z}^\top}{m}\right)}_{R(\boldsymbol{Z})} - \underbrace{\sum_{k=1}^{K}\frac{\operatorname{tr}(\boldsymbol{\Pi}_k)}{2m}\log\det\left(\boldsymbol{I} + \frac{d}{\varepsilon^2}\cdot\frac{\boldsymbol{Z}\boldsymbol{\Pi}_k\boldsymbol{Z}^\top}{\operatorname{tr}(\boldsymbol{\Pi}_k)}\right)}_{R^c(\boldsymbol{Z}\mid\boldsymbol{\Pi})}$$

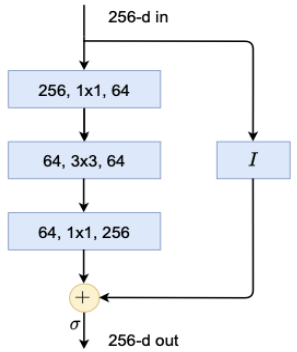
Unrolled optimization: gradient ascent on the MCR^2 objective,

$$\boldsymbol{Z}^{\ell+1} = \boldsymbol{Z}^{\ell} + \eta \cdot \frac{\partial \Delta R}{\partial \boldsymbol{Z}}\Big|_{\boldsymbol{Z}=\boldsymbol{Z}^{\ell}}$$





ReduNet



ResNet

ReduNet: A White-box Deep Network from the Principle of MCR^2 [YCY+**Ma**, JMLR2022].

How to Learn: Maximal Coding Rate Reduction (ReduNet)

Benign local and global optimization landscape of $\mathbf{MCR}^2$

Maximal Coding Rate Reduction (MCR^2)

$$\max_f \Delta R(\mathbf{Z}\,|\,\mathbf{\Pi}) = R(\mathbf{Z}) - R^c(\mathbf{Z}\,|\,\mathbf{\Pi})$$

Theorem [YCY+M NeurIPS2020].

The global optimal solution $\mathbf{Z}^\star = [\mathbf{Z}_1^\star, \mathbf{Z}_2^\star, \ldots, \mathbf{Z}_K^\star]$ of MCR^2 satisfies:

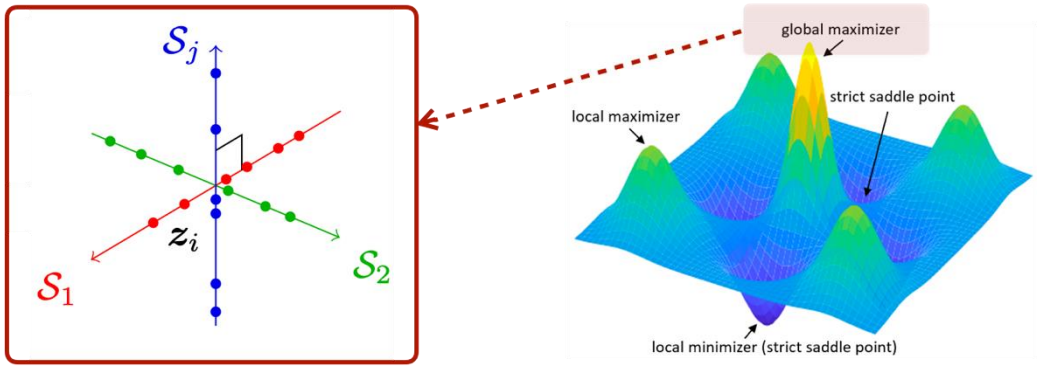
- Subspaces of different classes are orthogonal to each other, $(\mathbf{Z}_i^\star)^\top \mathbf{Z}_j^\star = \mathbf{0}$ for $i \neq j$;
- Each subspace achieves its maximal dimension, i.e., $\mathrm{rank}(\mathbf{Z}_k^\star) = d_k$, and and $d = \sum_{k=1}^{K} d_k$.

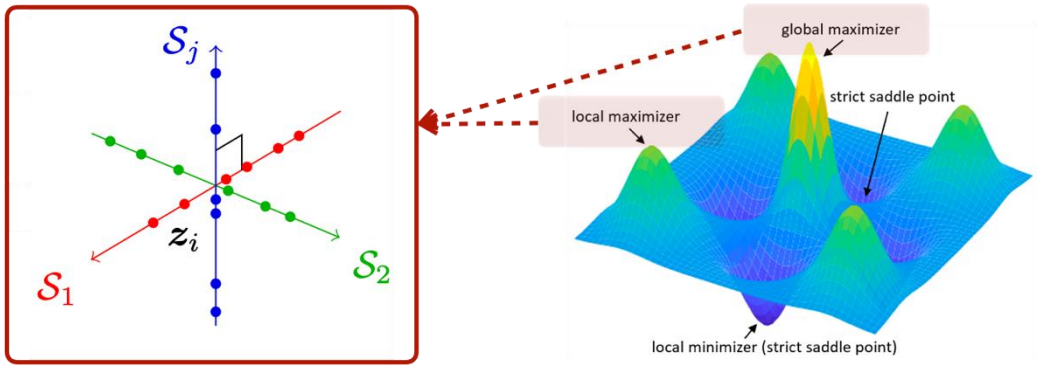
Regularized Maximal Coding Rate Reduction (MCR^2)

$$\max_f F(\mathbf{Z}) := \Delta R_\lambda(\mathbf{Z}\,|\,\mathbf{\Pi}) = R(\mathbf{Z}) - R^c(\mathbf{Z}\,|\,\mathbf{\Pi}) - \lambda \cdot \|\mathbf{Z}\|_F^2$$

Theorem [WLY+M 2024].

Every critical point $\{\mathbf{Z} : \nabla F(\mathbf{Z}) = \mathbf{0}\}$ is either a local maximizer or a strict saddle point: each local maximizer corresponds to a feature representation that consists of a family of orthogonal subspaces.



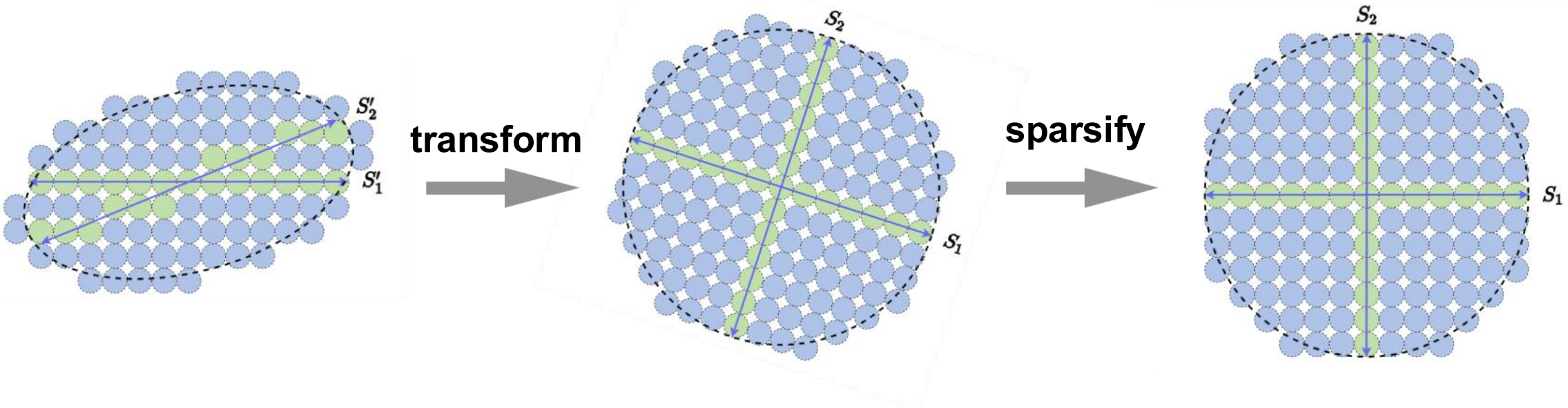


How to Learn: Sparse Rate Reduction (SRR)

Maximize the difference between the coding rate of all features and the coding rate of features within each subspace, and promote sparsity:

$$\max_{\boldsymbol{Z}} \ \mathrm{SRR}(\boldsymbol{Z} \mid \boldsymbol{U}_{[K]}) = R(\boldsymbol{Z}) - R^{c}(\boldsymbol{Z} \mid \boldsymbol{U}_{[K]}) - \lambda \|\boldsymbol{Z}\|_1$$

$-\lambda||\boldsymbol{Z}||_1$: measure how sparse all features are.



CRATE: White-box Transformer via Sparse Rate Reduction [YBP+**Ma,** JMLR2024].

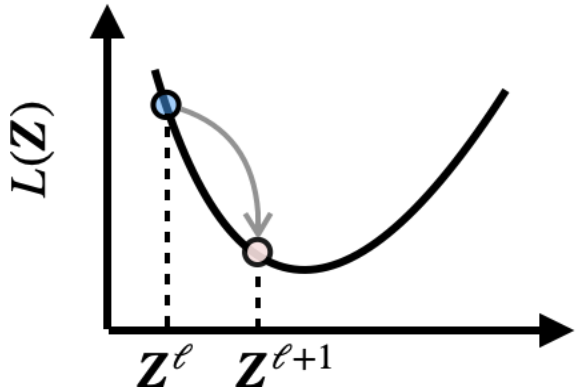
How to Learn: DNN to Realize Iterative Optimization

Optimize Sparse Rate Reduction via gradient descent (GD)

$$\min L_{SRR}(\boldsymbol{Z}) = -\mathrm{SRR}(\boldsymbol{Z} \mid \boldsymbol{U}_{[K]})$$

gradient descent

$$\boldsymbol{Z}^{\ell+1} = f^{\ell}(\boldsymbol{Z}^{\ell}) \approx \mathrm{Prox}[\boldsymbol{Z}^{\ell} - \eta \cdot \nabla L_{\mathrm{srr}}(\boldsymbol{Z}^{\ell})]$$



Sparse Rate Reduction (SRR):

$$\min_{f} L_{\mathrm{srr}}(\boldsymbol{Z} \mid \boldsymbol{U}_{[K]}) = \underbrace{R^{c}(\boldsymbol{Z} \mid \boldsymbol{U}_{[K]})}_{\text{compression}} + \underbrace{\lambda \|\boldsymbol{Z}\|_1 - R(\boldsymbol{Z})}_{\text{sparsification}}$$

$$\boldsymbol{Z}^{\ell+1} = \boldsymbol{Z}^{\ell} - \eta \cdot \nabla L(\boldsymbol{Z})\Big|_{\boldsymbol{Z}=\boldsymbol{Z}_{\ell}}$$

f^{ℓ}

Design the ℓ-th layer $f^{\ell} = f_2^{\ell} \circ f_1^{\ell}$ via an **Alternating Minimization Scheme**:

- Compression Step: $\boldsymbol{Z}^{\ell+1/2} = f_1^{\ell}(\boldsymbol{Z}^{\ell}) \approx \boldsymbol{Z}^{\ell} - \eta \cdot \nabla R^{c}(\boldsymbol{Z}^{\ell}; \boldsymbol{U}_{[K]})$;
- Sparsification Step: $\boldsymbol{Z}^{\ell+1} = f_2^{\ell}(\boldsymbol{Z}^{\ell+1/2}) \approx \mathrm{Prox}_{\lambda\|\cdot\|_1}\left[\boldsymbol{Z}^{\ell+1/2} - \eta \cdot \mathrm{grad}(\boldsymbol{Z}^{\ell+1/2})\right]$.

CRATE: White-box Transformer via Sparse Rate Reduction [YBP+**Ma,** JMLR2024].

How to Learn: Interpretation of Each Layer

Gradient Descent (GD) operator:

Sparsification Step:

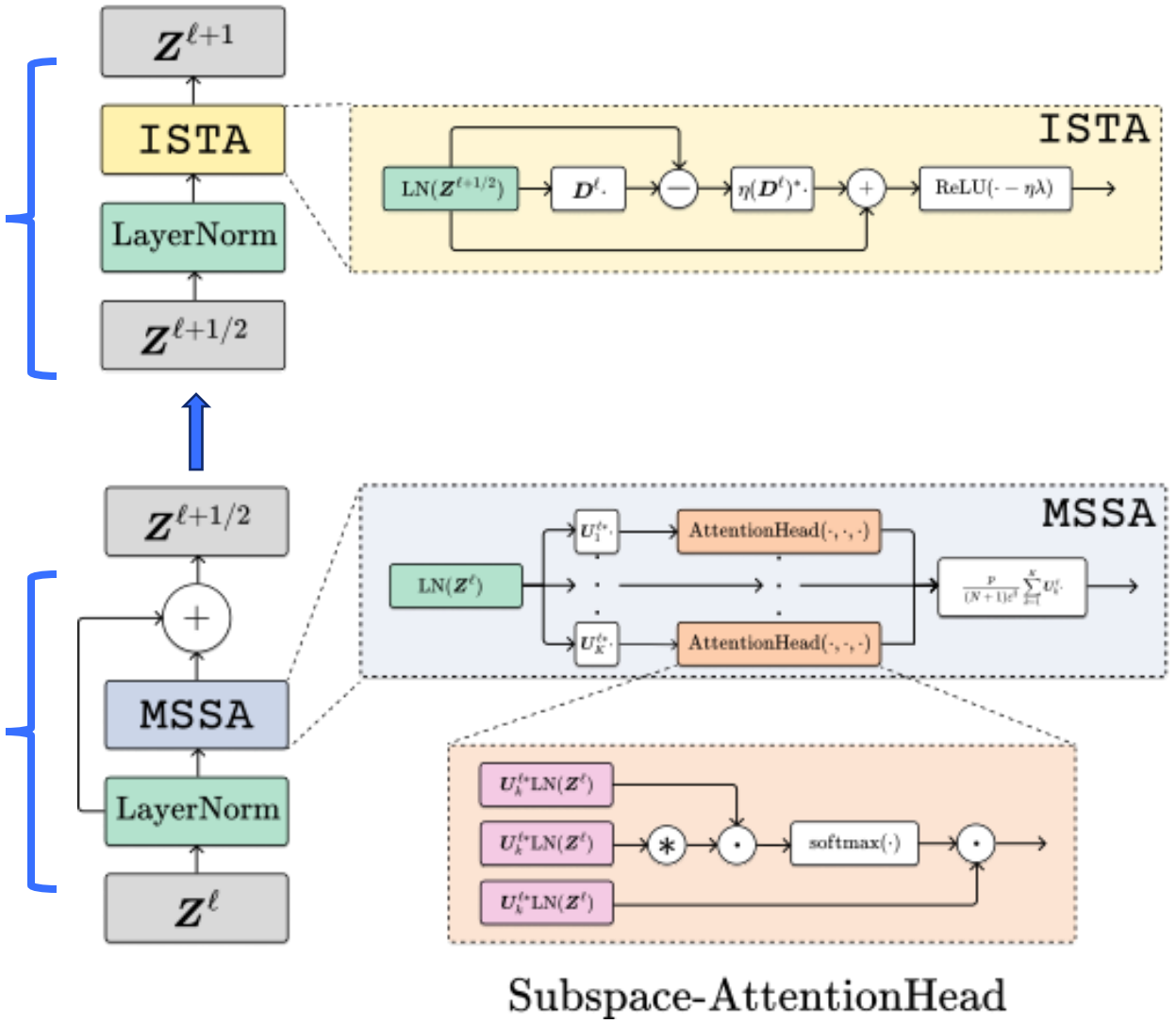
$$\boldsymbol{Z}^{\ell+1} = \mathsf{Prox}_{\lambda\|\cdot\|_1}\left[\boldsymbol{Z}^{\ell+1/2} - \eta \cdot \mathsf{grad}(\boldsymbol{Z}^{\ell+1/2})\right]$$

$$= \underbrace{\mathsf{ReLU}\left(\boldsymbol{Z}^{\ell+1/2} + \eta \cdot \boldsymbol{D}^\top(\boldsymbol{Z}^{\ell+1/2} - \boldsymbol{D}\boldsymbol{Z}^{\ell+1/2}) - \eta \cdot \lambda \cdot \boldsymbol{1}\right)}_{\mathsf{ISTA}(\boldsymbol{Z}^{\ell+1/2})}$$

Compression Step:

$$\boldsymbol{Z}^{\ell+1/2} = \boldsymbol{Z}^{\ell} - \eta \cdot \nabla R^c(\boldsymbol{Z}^{\ell}; \boldsymbol{U}_{[K]})$$

$$\approx \left(1 - \eta \cdot \beta\right)\boldsymbol{Z}^{\ell} + \eta \cdot \beta \cdot \mathsf{MSSA}(\boldsymbol{Z}^{\ell} \mid \boldsymbol{U}_{[K]})$$

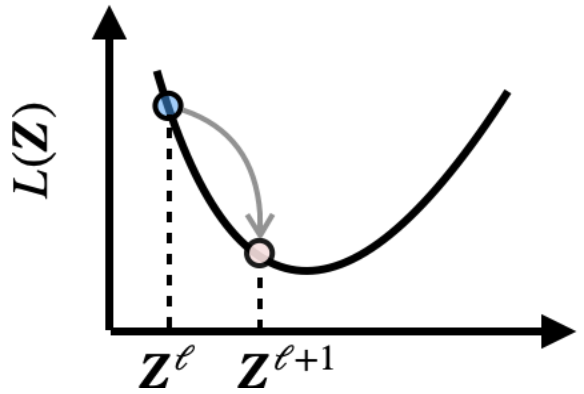


CRATE: White-box Transformer via Sparse Rate Reduction [YBP+**Ma,** JMLR2024].

How to Learn: DNN to Realize Iterative Optimization

Optimize Sparse Rate Reduction via iterative gradient descent (GD)

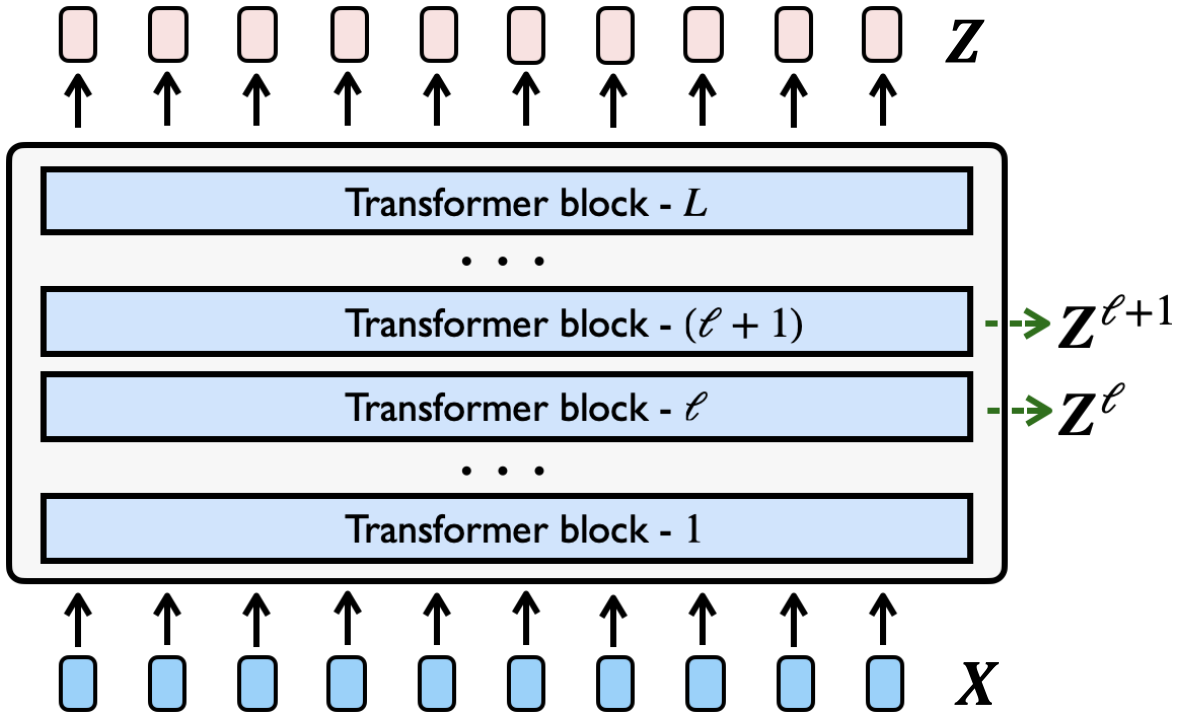
$$\min L_{SRR}(\boldsymbol{Z}) = -\mathrm{SRR}(\boldsymbol{Z} \mid \boldsymbol{U}_{[K]})$$



$$\boldsymbol{Z}^{\ell+1} = \boldsymbol{Z}^{\ell} - \eta \cdot \nabla L(\boldsymbol{Z})\Big|_{\boldsymbol{Z}=\boldsymbol{Z}_{\ell}}$$

$$f^{\ell}$$

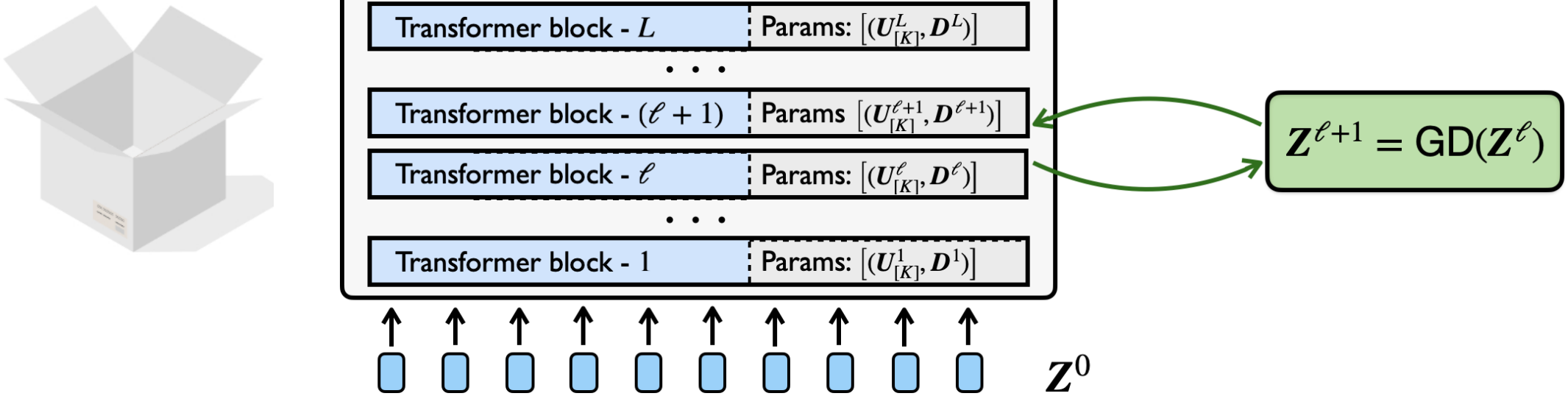
$$f : \boldsymbol{X} = \boldsymbol{Z}^0 \xrightarrow{f^1} \boldsymbol{Z}^1 \xrightarrow{f^2} \boldsymbol{Z}^2 \longrightarrow \cdots \xrightarrow{f^L} \boldsymbol{Z}^L = \boldsymbol{Z}$$



Each layer of a deep network (e.g., Transformer) realizes a GD operator.

How to Learn: Interpretation of the Whole Network

- Forward encoding: given fixed subspaces and dictionaries $(\boldsymbol{U}^{\ell}_{[K]}, \boldsymbol{D}^{\ell})_{\ell \in [L]}$, each layer performs compression and sparsification on representations.
- Backward learning the "codebook": backpropagation to learn subspaces and dictionaries $(\boldsymbol{U}^{\ell}_{[K]}, \boldsymbol{D}^{\ell})_{\ell \in [L]}$ from data.

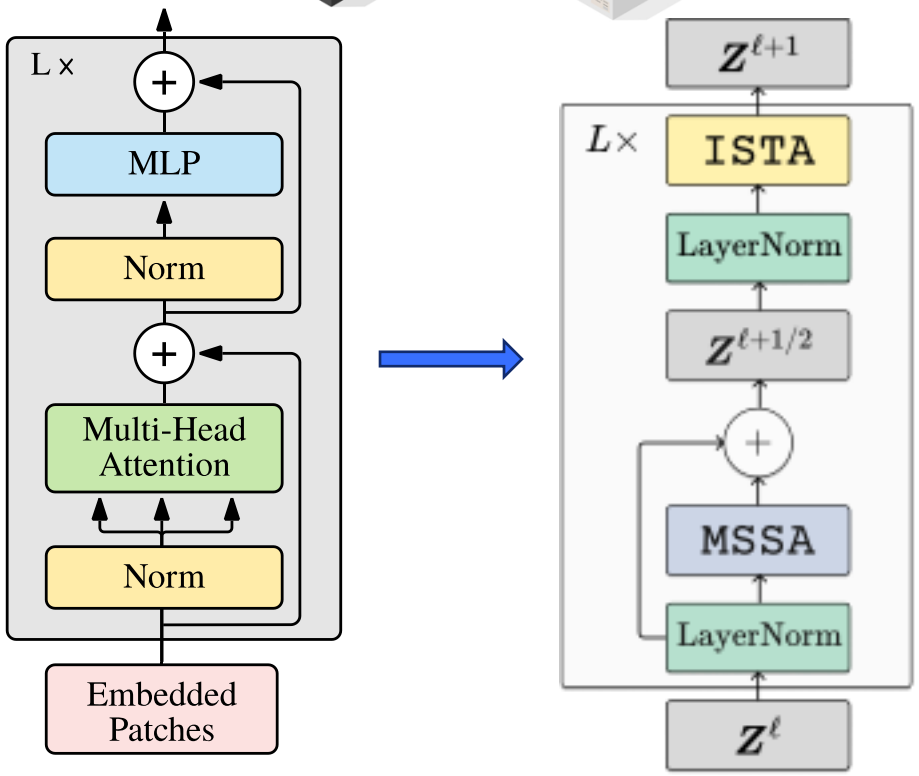


CRATE: White-box Transformer via Sparse Rate Reduction [YBP+**Ma,** JMLR2024].

How to Learn: Better Semantic Interpretability

Not only mathematically fully explainable, but also semantically more interpretable!

Transformer → CRATE [NeurIPS 2023]



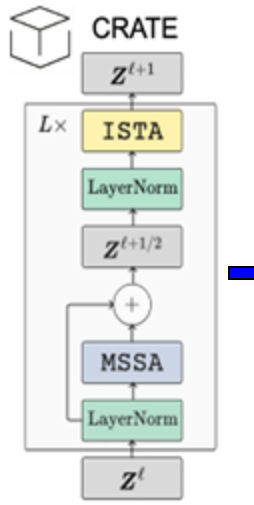
For real data (ImageNet), CRATE learns semantically meaningful structures.

How to Learn: Better Networks from First Principles

No more trial and error to design better network architectures:

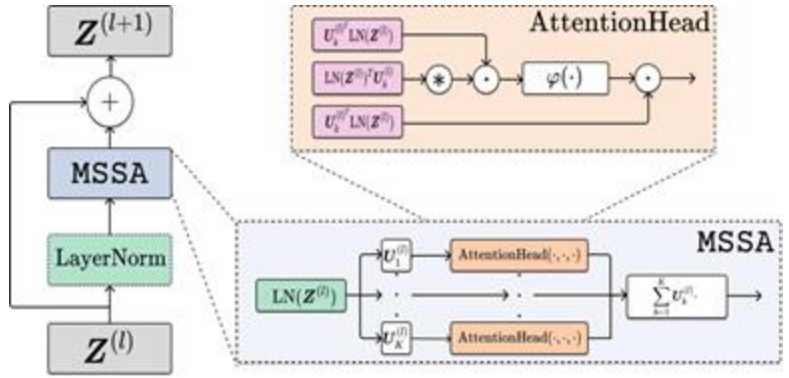
- Explainable [NeurIPS 2023]
- Scalable [NeurIPS 2024]
- More efficient [ICLR 2025]
- More compact [CPAL 2025]

NeurIPS 2023

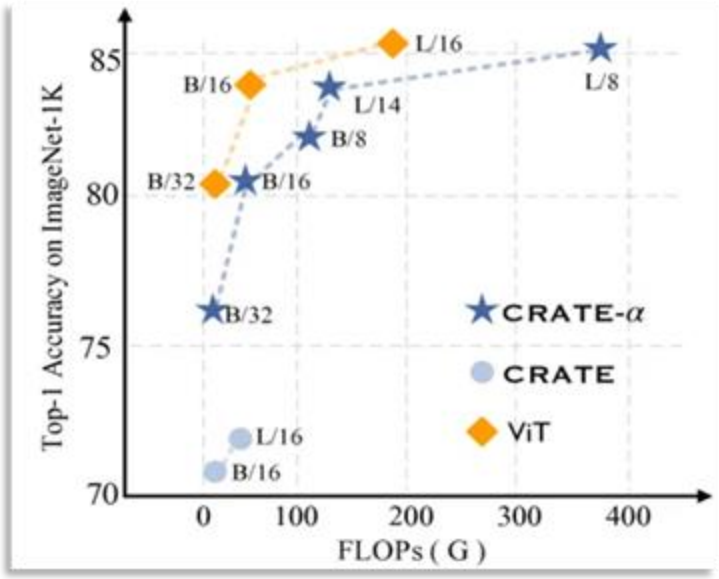


More Compact

(Attention Only, ICML2025)

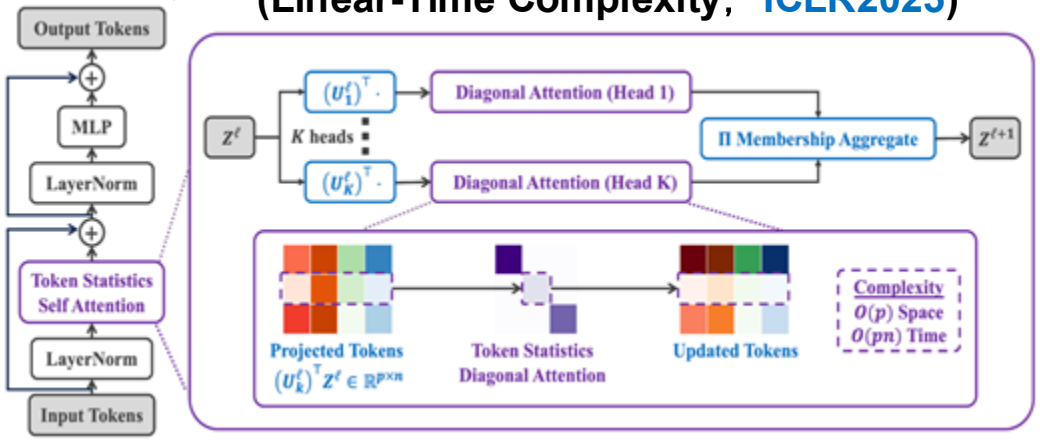


Scalable, **NeurIPS 2024**



More Efficient

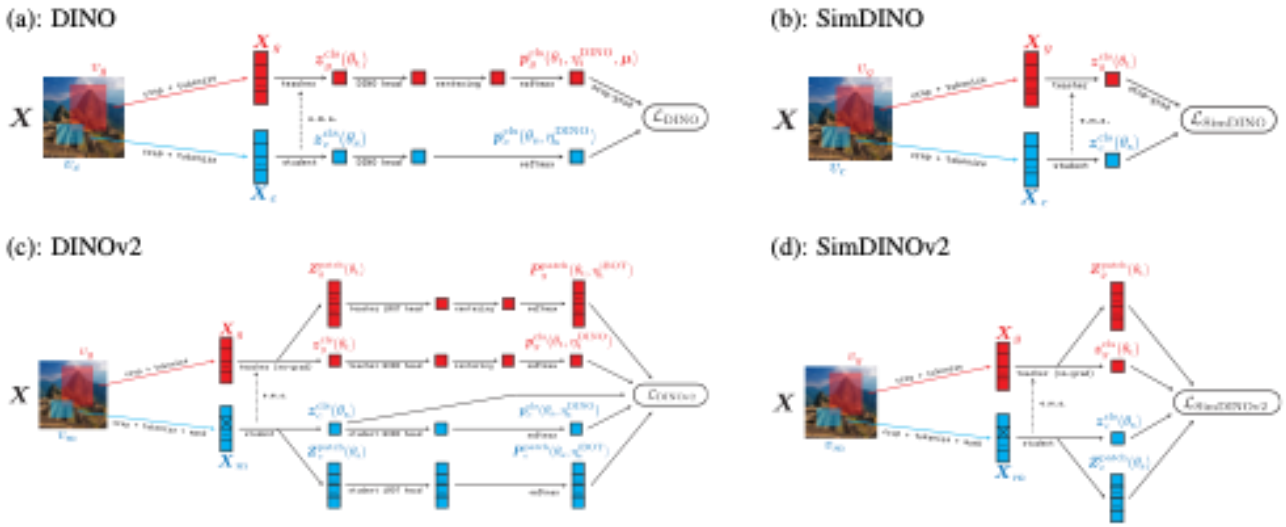
(Linear-Time Complexity, ICLR2025)



How to Learn: Better Networks from First Principles

No more trial and error to design better network architectures:

SimDINO: Simplifying DINO via Coding Rate Regularization [ICML2025]



Classification (ImageNet)

Method	Model	Epochs	*k*-NN	Linear
DINO	ViT-B	100	72.9	76.3
SimDINO	ViT-B	100	74.9	77.3
DINO	ViT-L	100	–	–
SimDINO	ViT-L	100	75.6	77.4
DINOv2	ViT-B	100	76.0	77.2
SimDINOv2	ViT-B	100	78.1	79.7
DINOv2	ViT-L	100	80.8	82.0
SimDINOv2	ViT-L	100	81.1	82.4
SwAV	ViT-S	800	66.3	73.5
MoCov3	ViT-B	300	–	76.7

Segmentation (Microsoft COCO)

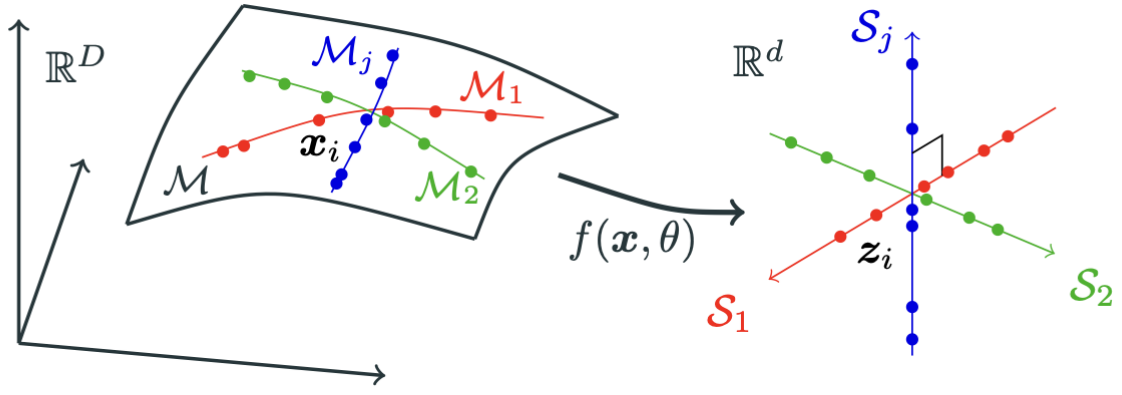
		Detection ↑			Segmentation ↑		
Method	Model	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}	AP
SimDINO	ViT-L/16	**5.4**	1.9	2.4	4.5	1.4	1.9
SimDINO	ViT-B/16	5.2	**2.0**	**2.5**	**4.7**	**1.5**	**2.0**
DINO	ViT-B/16	3.9	1.5	1.8	3.1	1.0	1.4
DINO	ViT-B/8	5.1	2.3	2.5	4.1	1.3	1.8

Hyperparameter		SimDINOv2	DINOv2	SimDINO	DINO
Model	Patch size	16			
	Register tokens	4		0	
	Pos-embedding anti-alias	True		False	
	Init layer scale	0.1	1e-5	-	
	Drop path rate	0.3		0.1	
	Weight normalize last layer	removed	True	removed	True
	Output prototypes K	removed	65536	removed	65536
Pipeline	Init EMA momentum	0.9	0.992	0.996	
	Centering temperature	removed	0.07	removed	0.07
	Warm-up temperature	removed	0.04	removed	0.04
	Warm-up temperature epochs	removed	30	removed	30
	iBOT sample prob.	0.5		-	
	iBOT mask ratio	0.1-0.5		-	
	iBOT head tying	False		-	
	Koleo loss weight	removed	0.1	-	
Data	Global crops scale	0.4 - 1			
	Local crops scale	0.05 - 0.4			
	Local crops number	10			
	Global crops size	224			
	Local crops size	96			
Optim.	Batch size	128x8		64x8	
	Epochs	100			
	Warm-up epochs	10			
	Freeze last layer epochs	removed	1	removed	1
	Learning rate	0.004		0.002	
	Layerwise lr decay	0.9		-	
	Weight decay	0.04			
	Weight decay end	0.4			
	Gradient clip	3.0		0.3	

How to Learn: Summary with a Comparison

No more trial and error to design better network architectures:

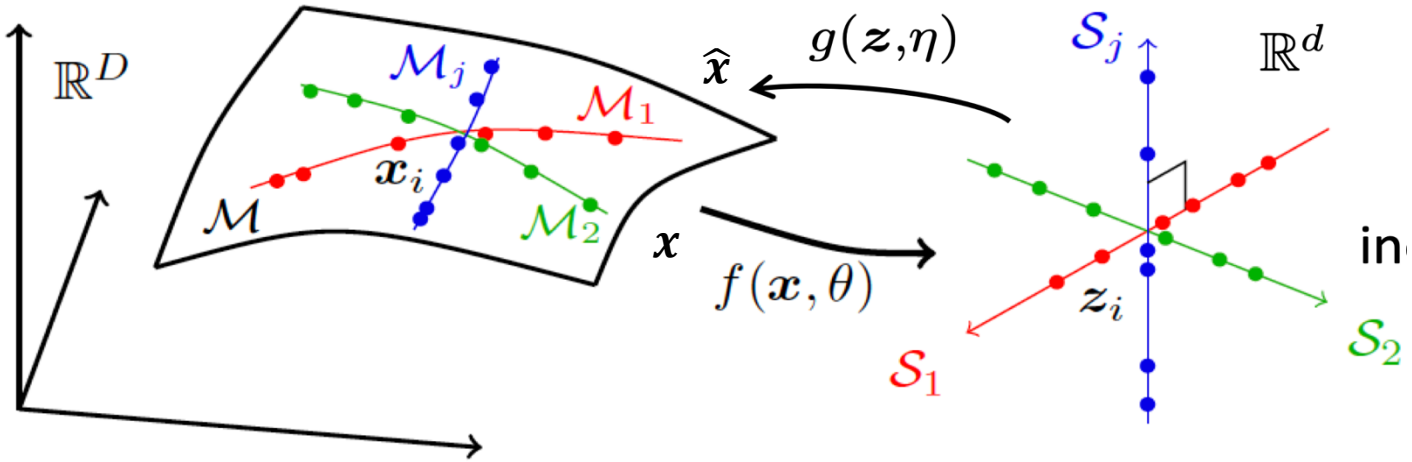
Deep Representation Learning



	Black-box	White-box
Objective	Input/label fitting	Information gain
Deep architecture	Empirical design	Iterative optimization
Representation	Hidden/latent	Incoherent subspaces/ dictionaries

Why Correct? (Consistency)

Bi-directional encoding and decoding (e.g., compression and generation)

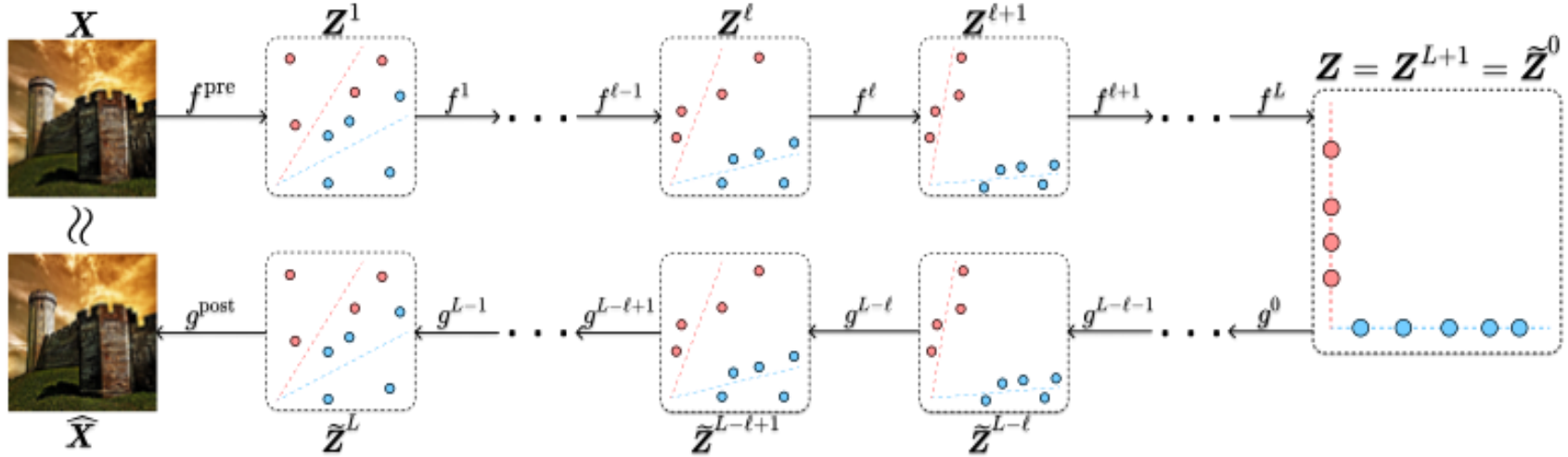


in

$p(\hat{\boldsymbol{x}}) \approx p(\boldsymbol{x})$? or $\hat{\boldsymbol{X}} \approx \boldsymbol{X}$?

Autoencoding

(2024 Nobel Prize Physics)



Why Correct? (Consistency)

Bi-directional encoding and decoding (e.g., compression and generation)

Masked autoencoding with a **whitebox architecture**
[PBWY+**Ma**, ICLR2024]

$$\mathcal{P}_{\Omega}(\boldsymbol{x}) \;\rightarrow\; \widehat{\boldsymbol{x}} \approx \boldsymbol{x}$$

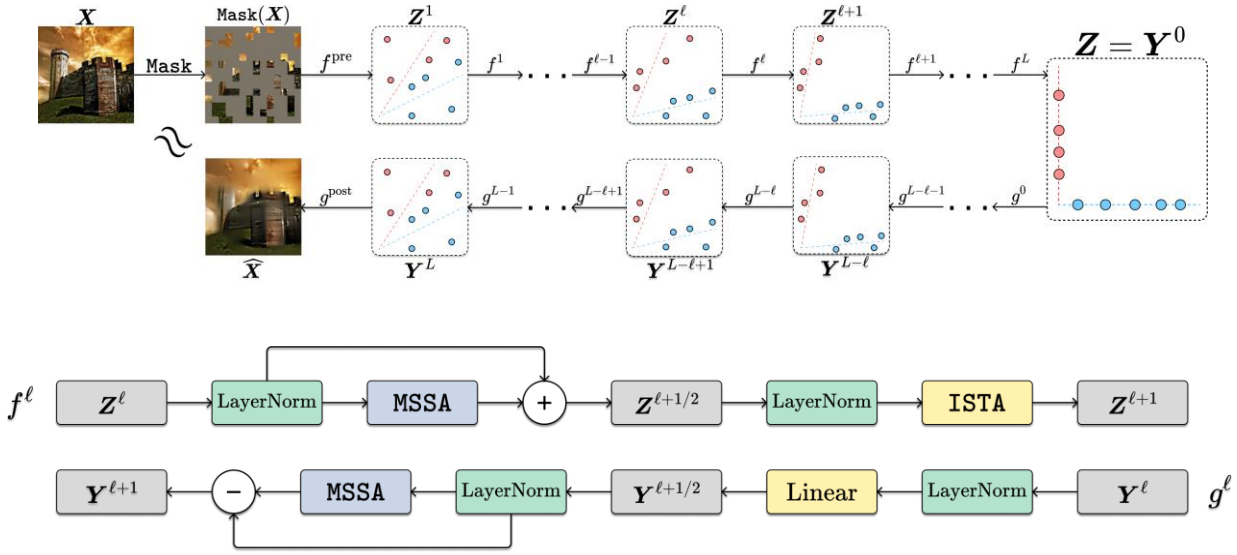
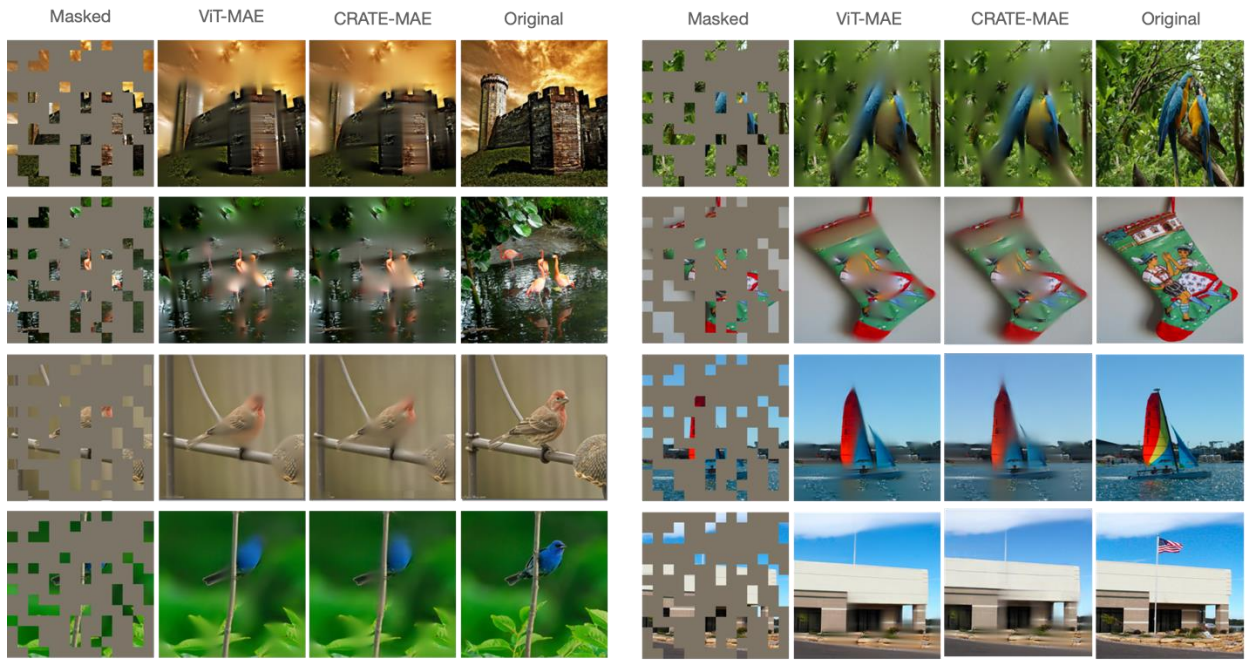
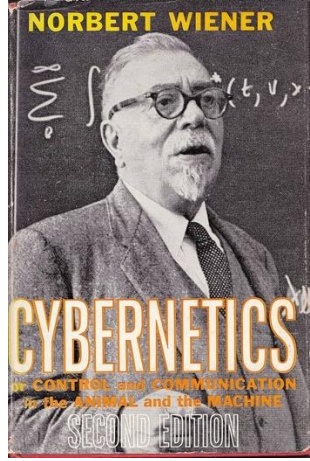


Image completion with 75% patches/pixels masked

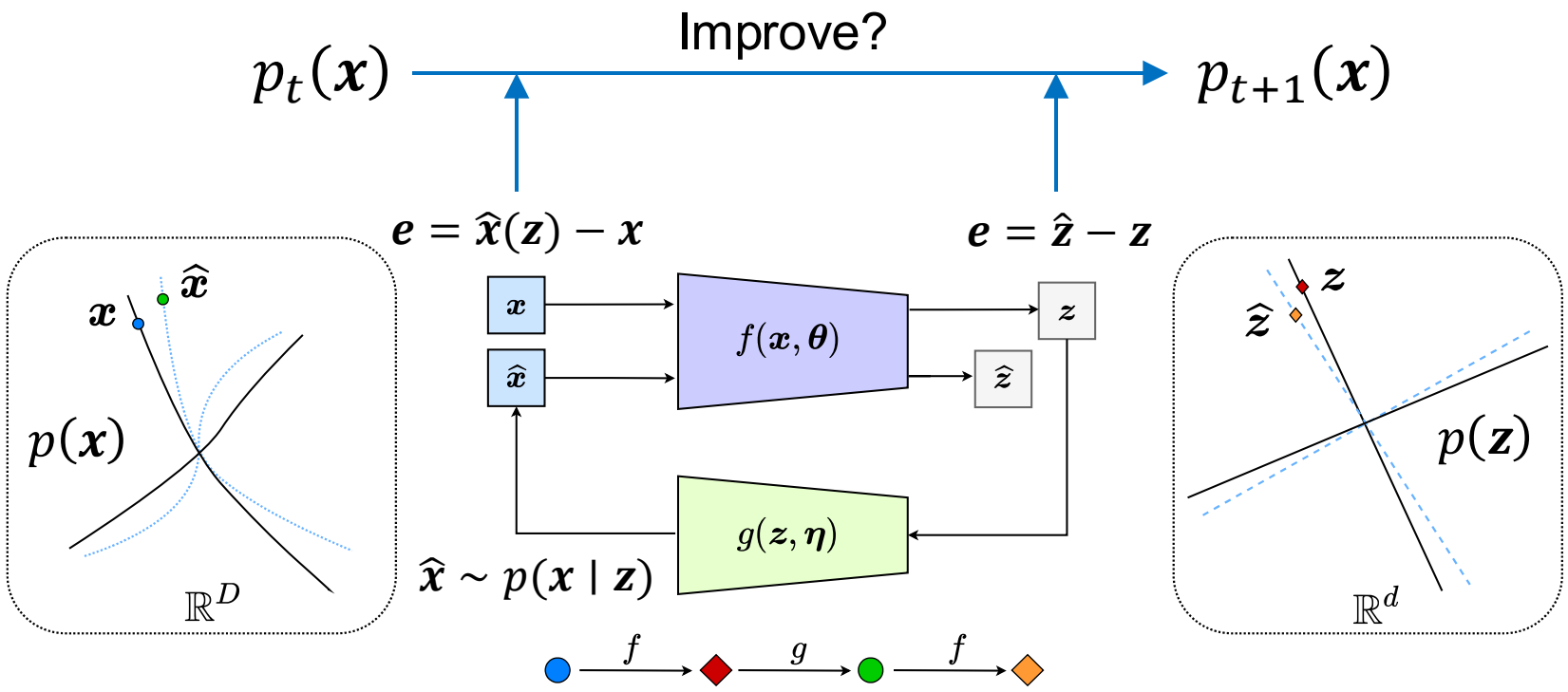


Models: CRATE-MAE-Base versus ViT-MAE-Base [PBWY**M**, ICLR2024]

Towards Autonomous Intelligence (AI 2.0)



How to self-learn a more consistent representation, continuously?



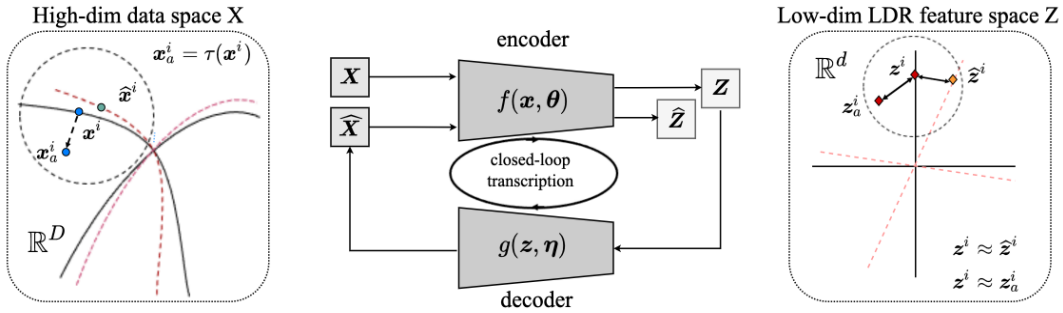
In nature, all intelligent systems learn from closed-loop feedback! (Cybernetics)

Towards AI 2.0: Close the Loop via Minimax Game

Closed-loop systems learnt via minimax game do not forget catastrophically!

Incremental Learning via Closed-Loop [TDWLY+**Ma**, ICLR 2023]

Unsupervised Learning of Structured Memory: one sample at a time[11]



$$\max_{\checkmark} \min_{\not\Phi} \; R(Z) + \Delta R(Z, \hat{Z})$$

$$\text{subject to} \quad \sum_{i \in N} \Delta R(z^i, \hat{z}^i) = 0, \quad \text{and} \quad \sum_{i \in N} \Delta R(z^i, z_a^i) = 0.$$

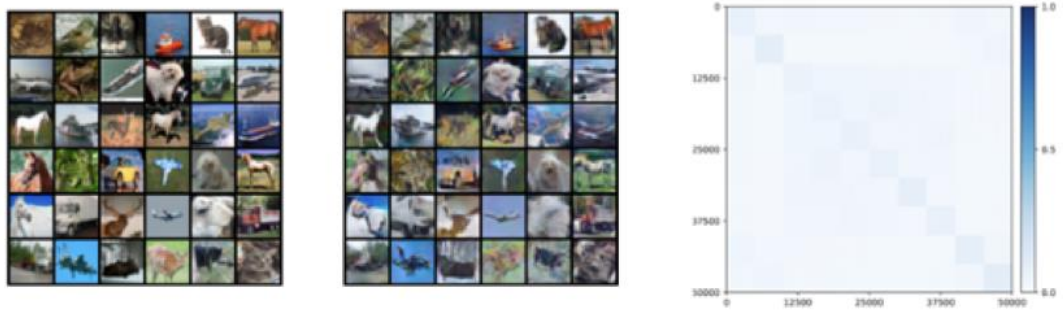


Figure: Sample-wise self-consistency and block-diagonal structures.

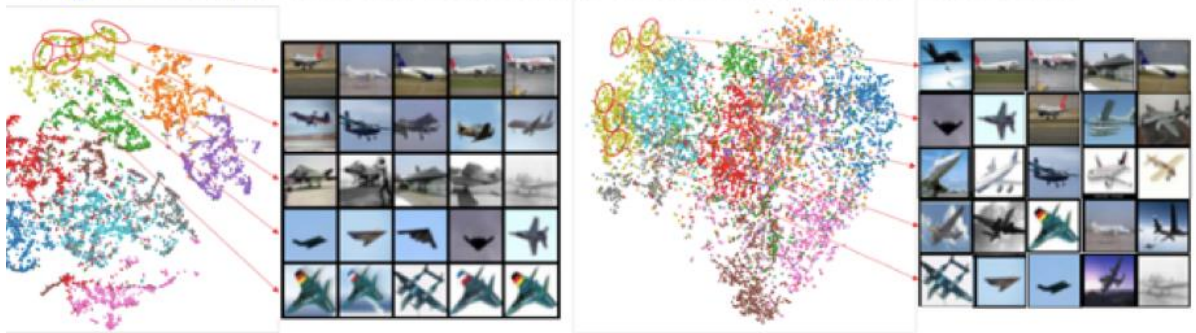


Figure: t-SNE of learned features . Left: U-CTRL and Right: MoCoV2.

EMP-SSL: Towards Self-Supervised Learning in One Training Epoch, Shengbang Tong, Yubei Chen, Yi Ma, Yann Lecun, arXiv:2304.03977

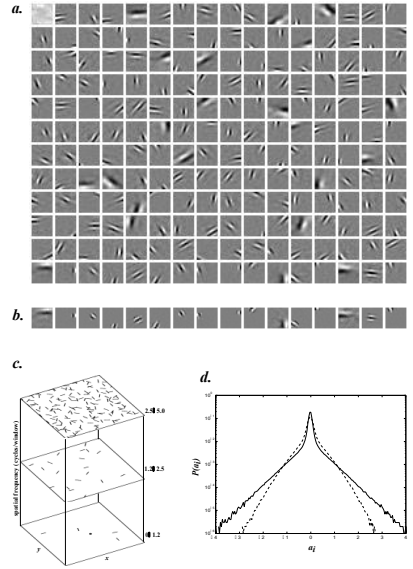
[11] Unsupervised Learning of Structured Representations via Closed-Loop Transcription, S. Tong, Yann LeCun, and Yi Ma, arXiv:2210.16782, 2022.

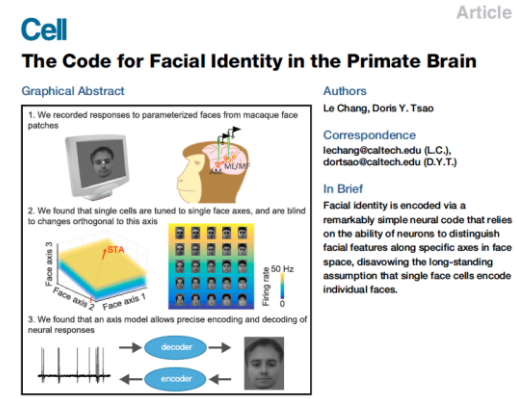
Towards AI 2.0: Time to Learn from Nature Again?

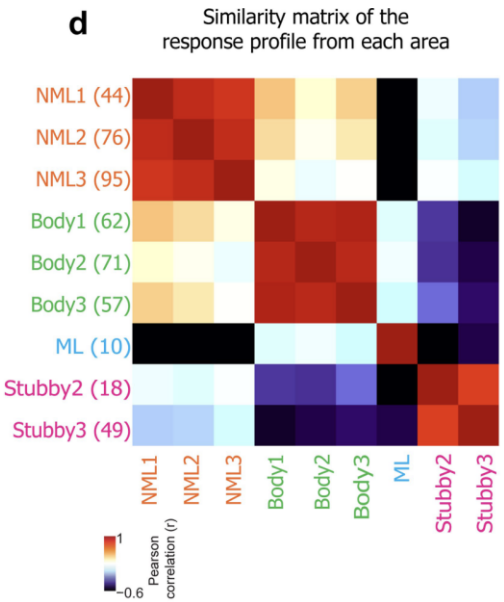
Similar characteristics and mechanisms are ubiquitous in nature!

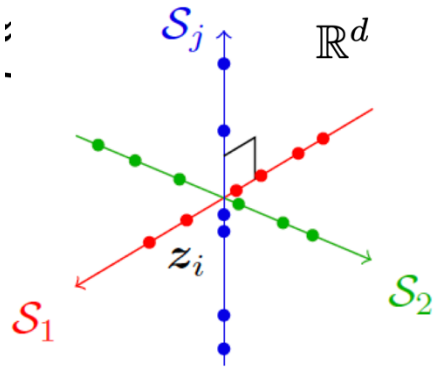
- Sparse coding in visual cortex (Olshausen, Nature 1996) [12].
- Subspace embedding (Tsao, Cell 2017, Nature 2020).[13]
- Predictive coding in visual cortex (Rao, Nature Neuroscience 1999).

sparse coding in visual cortex









Towards AI 2.0 (Neural Science)

A position paper about Intelligence in 2022:

On the Principles of Parsimony and Self-Consistency for the Emergence of Intelligence

Yi Ma[†‡1], Doris Tsao[†2], Heung-Yeung Shum[†3]

[1]*Electrical Engineering and Computer Science Department, University of California, Berkeley, CA 94720, USA*

[2]*Department of Molecular & Cell Biology and Howard Hughes Medical Institute, University of California, Berkeley, CA 94720, USA*

[3]*International Digital Economy Academy, Shenzhen 518045, China*

[†]E-mail: yima@eecs.berkeley.edu; dortsao@berkeley.edu; hshum@idea.edu.cn

Ma+TS, FITEE 2022

Towards AI 2.0 (Neural Science)

- **Parsimony:** what's in neuroscience to verify this principle?
- **Self-consistency:** what's in neuroscience to verify this principle?
- **Forward optimization** versus backward propagation?
- **Closed-loop** versus open-loop?
- **Self-correcting** or self-improving mechanisms?

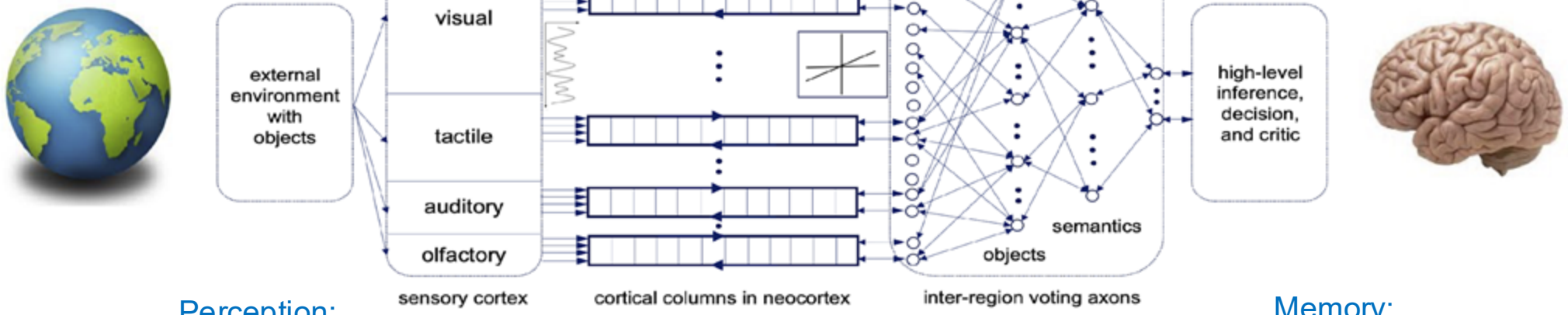
Towards AI 2.0: How to Implement (Computer Science)

To understand intelligence, one must understand computational complexity:

Incomputable ⟹ computable ⟹ tractable ⟹ scallable ⟹ natural

Kolmogorov complexity	Turing & Shannon	NP vs P	DNN & BP	Closed-loop & feedback?

A massive parallel, distributed, hierarchical system of autoencoders (**cortical columns in cortex**)

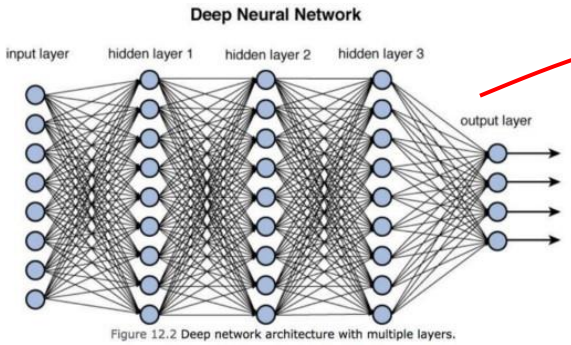


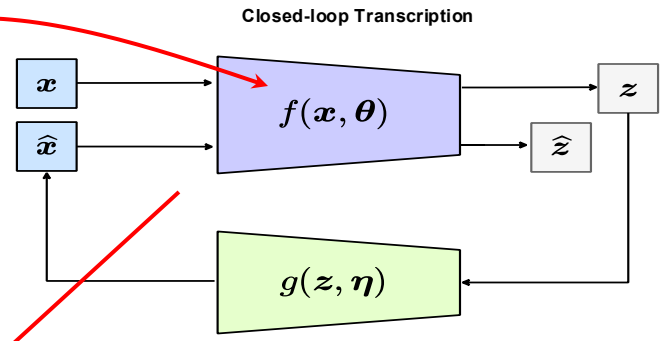
Perception:
vision-centric multi-modal senses

Memory:
language-centric knowledge representation

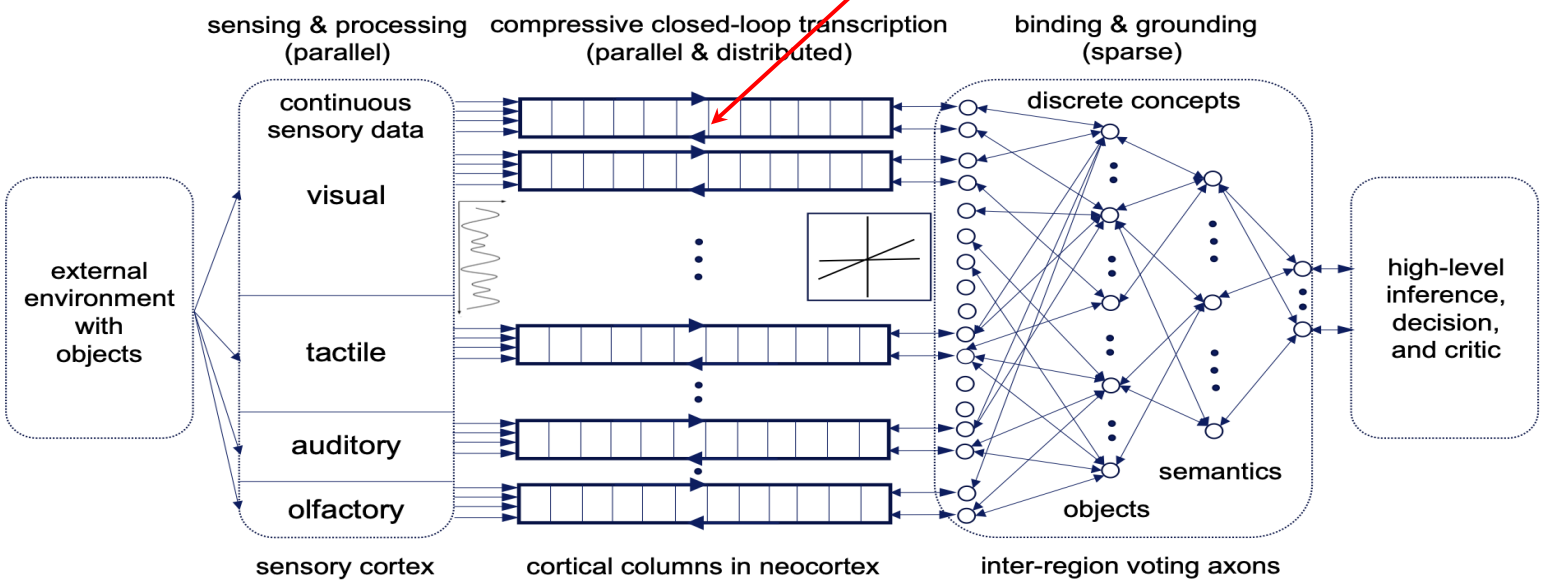
Towards AI 2.0: Time to Learn from Nature Again?

Neural networks are nature's optimization algorithms that maximize information gain.

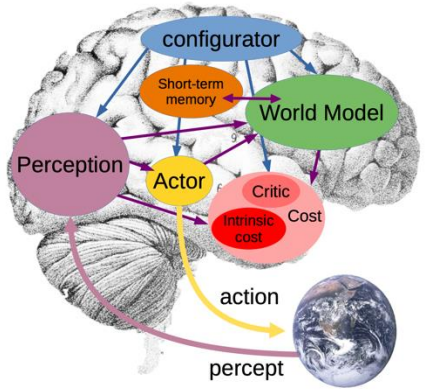




Closed-loop transcription is a basic unit for autonomously learn consistent knowledge.



Parallel, distributed, hierarchical autoencoders efficiently learn knowledge of the external world.



A unified purpose of intelligence: maximize "information gain" with every unit, at every stage!

What is Intelligence?

Definition [Intelligence]: an intelligent system is one that has the mechanisms for **self-correcting** and **self-improving** its existing knowledge (or information).

$$\text{Knowledge} = \int_0^t \text{Intelligence},$$

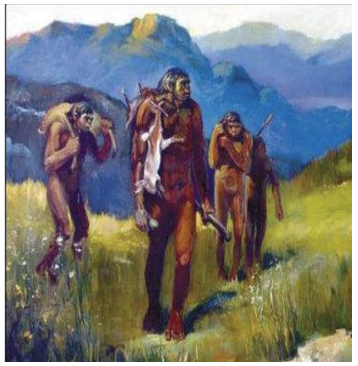
$$\text{Intelligence} = \frac{d}{dt}\text{Knowledge}.$$

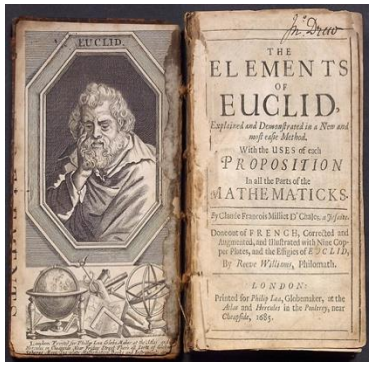
Any system without such mechanisms, however large, does not have any intelligence!

vs

Who has intelligence, who has knowledge?

Evolution of Intelligence in Nature: Four Stages





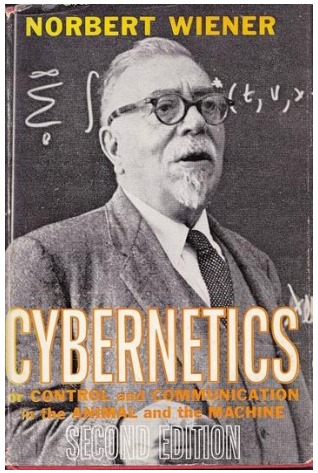
Phylogenetic **Ontogenetic** **Societal** **Artificial**

Intelligence is all about how to encode and improve information for better prediction of the world!

	Phylogentic	Ontogenetic	Societal	Artificial
Codebook	Amino Acids	Neural Networks	Alphabet & Words	Mathematics/Logic
Information	Genes/DNAs	Memory	Natural Languages	Scientific Facts
Improvement	Natural Selection	Continuous Feedback	Trial & Error	Hypothesis Testing

Today's "Artificial Intelligence" is not that Artificial Intelligence!

A quote from the 1956 Dartmouth proposal: "*An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves*".



1940s (animal intelligence)

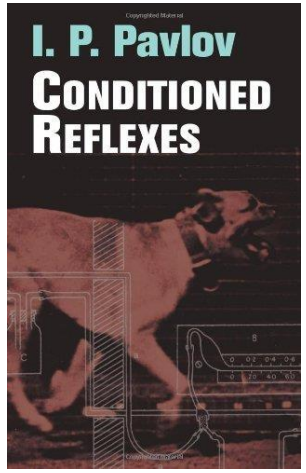
- Signal processing
- Information Rep.
- Prediction
- Error correction
- Optimal control
- Game theory

1956 (unique to human)

- Abstraction
- Logic deduction
- Causality
- Hypothesis forming & testing
- Problem solving

Today's AI (animal or human?)

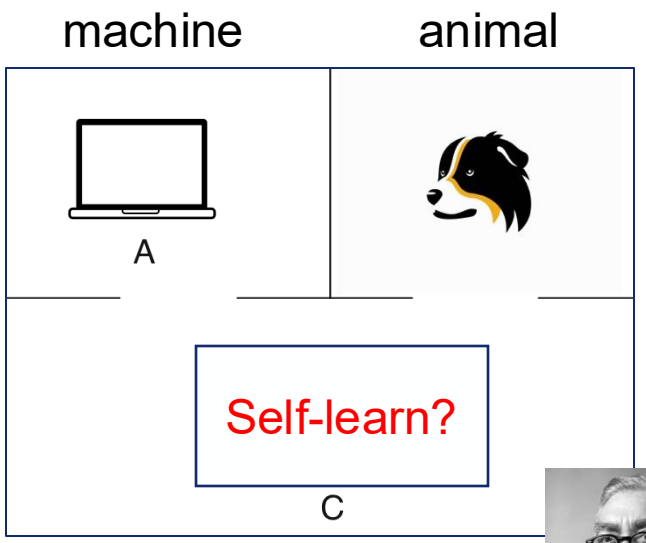
- Denoising
- Compression
- Object recognition
- Image generation
- Text generation
- Reinforce learning



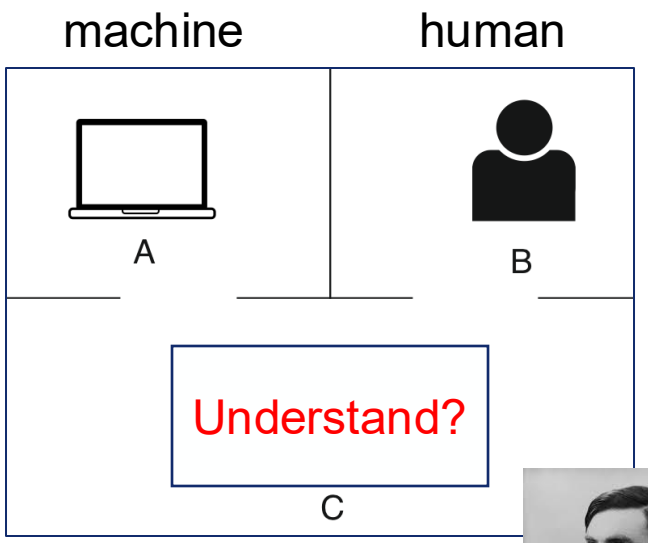
An Open Question: Scientific Tests for Intelligence?

How to scientifically certify different types of ability of an "intelligent" system:

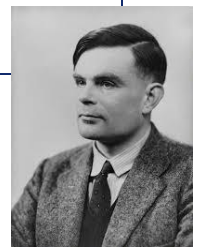
- **Memorize:** simply having memorized a large amount of knowledge-carrying data and regenerating them;
- **Self-Learn**: being able to autonomously and continuously develop better knowledge from new observations;
- **Understand**: having truly understood existing knowledge and knowing how to deduce and apply it correctly;
- **Theorize**: being able to generate new scientific hypotheses and mathematical theories and verify them.

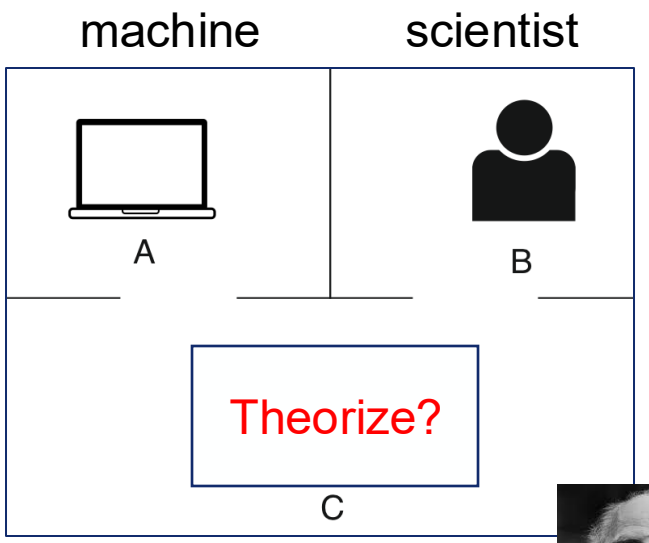


Wiener Test

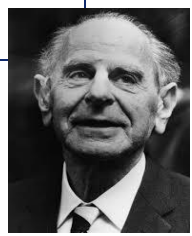


Turing Test





Popper Test



Epilogue

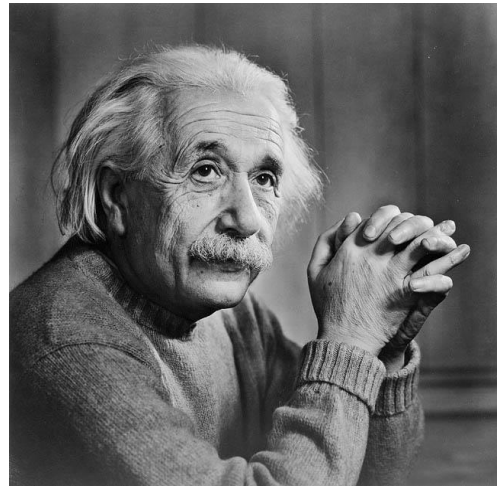
Seek a scientific and theoretical foundation for Intelligence:

- **what to learn?** ***parsimony***
- **how to learn?** ***compression***
- **why correct?** ***consistency***

大道至简

"Everything should be made as simple as possible,
but not any simpler."

-- Albert Einstein



Epilogue

Coming soon: A new textbook in 2025

Learning Deep Representations of Data Distributions

Yi Ma, University of Hong Kong

Druv Pai, University of California, Berkeley

Sam Buchanan, Toyota Technological Institute at Chicago

Peng Wang, University of Macau

with contributions from:

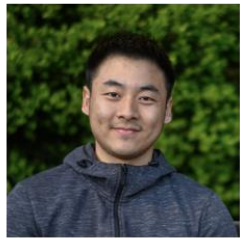
Yaodong Yu, University of Maryland, College Park

Acknowledgement

A Truly Multi-University and Multi-Disciplinary Effort from Academia!

Druv Pai
UC Berkeley

Sam Buchanan
TTI Chicago



Yaodong Yu
OpenAI, U. Maryland

Ziyang Wu
UC Berkeley

Shengbang Tong
New York University

Tianzhe Chu
Hong Kong University

Benjamin Haeffele
Johns Hopkins

Peng Wang
UMichigan

Simon Zhai
UC Berkeley, DeepMind

Jack Bai
UIUC

Ryan Chan
UPenn

NSF

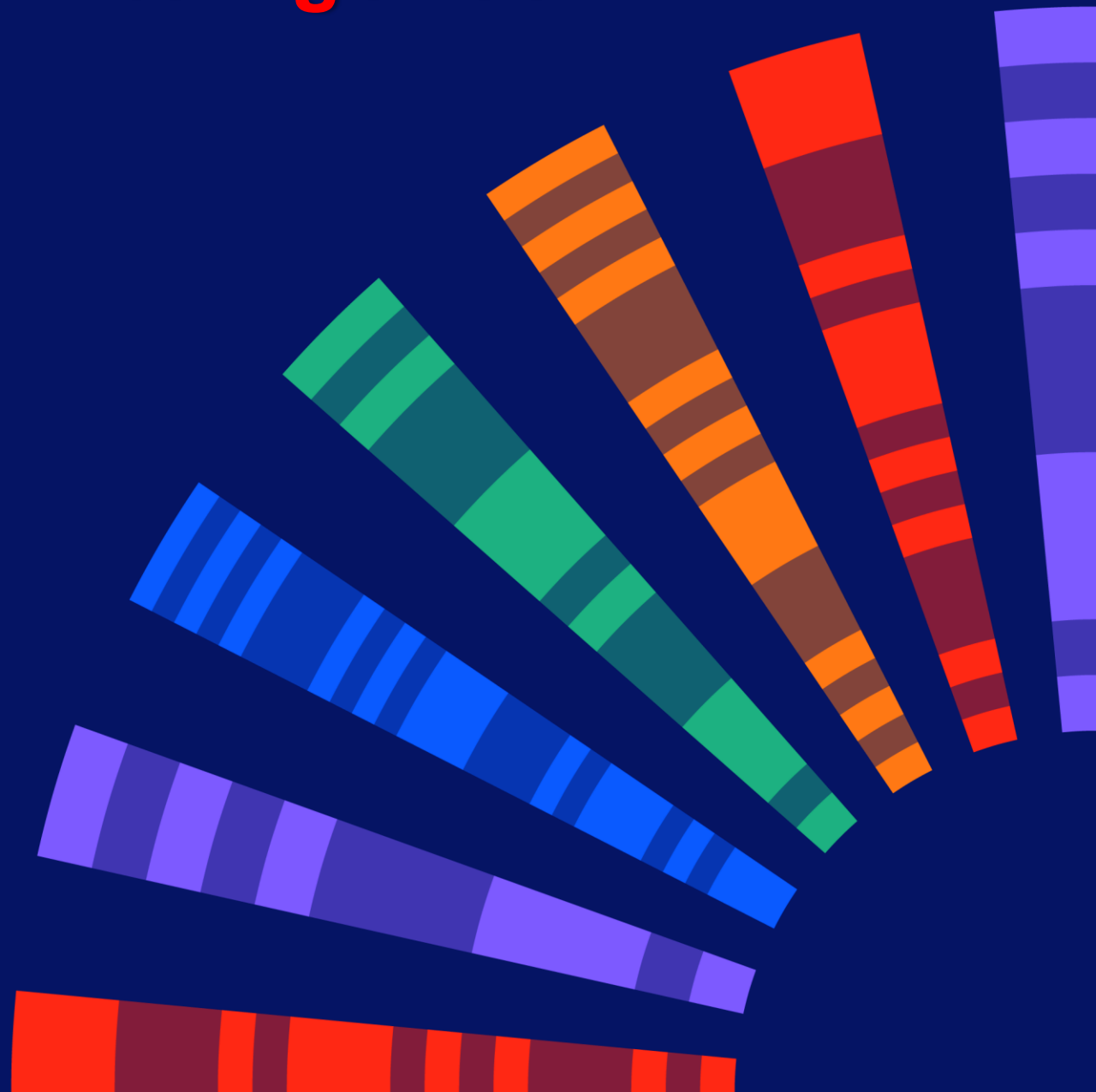
SIMONS FOUNDATION

HKU Musketeers Foundation
Institute of Data Science
香港大學同心基金數據科學研究院

Engram

Pursuing the Nature of Intelligence

Thanks



SCHOOL OF
COMPUTING &
DATA SCIENCE
The University of Hong Kong