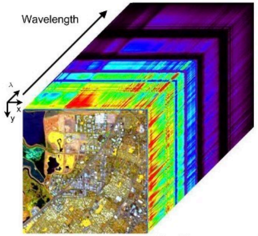
From Shallow to Deep Representation Learning in Imaging and Beyond: Global Nonconvex Theory and Algorithms

Qing Qu

Dept. of EECS, University of Michigan

September 7, 2021

Data Increasingly Massive & High-Dimensional...



hyperspectral imaging

autonomous driving

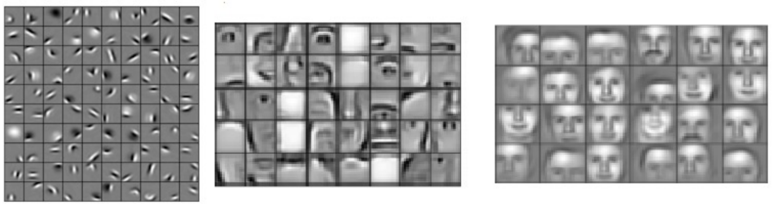
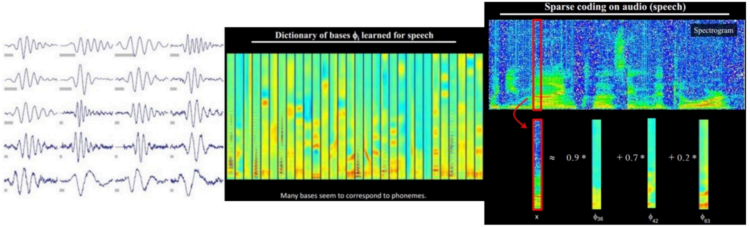


image representations

social network

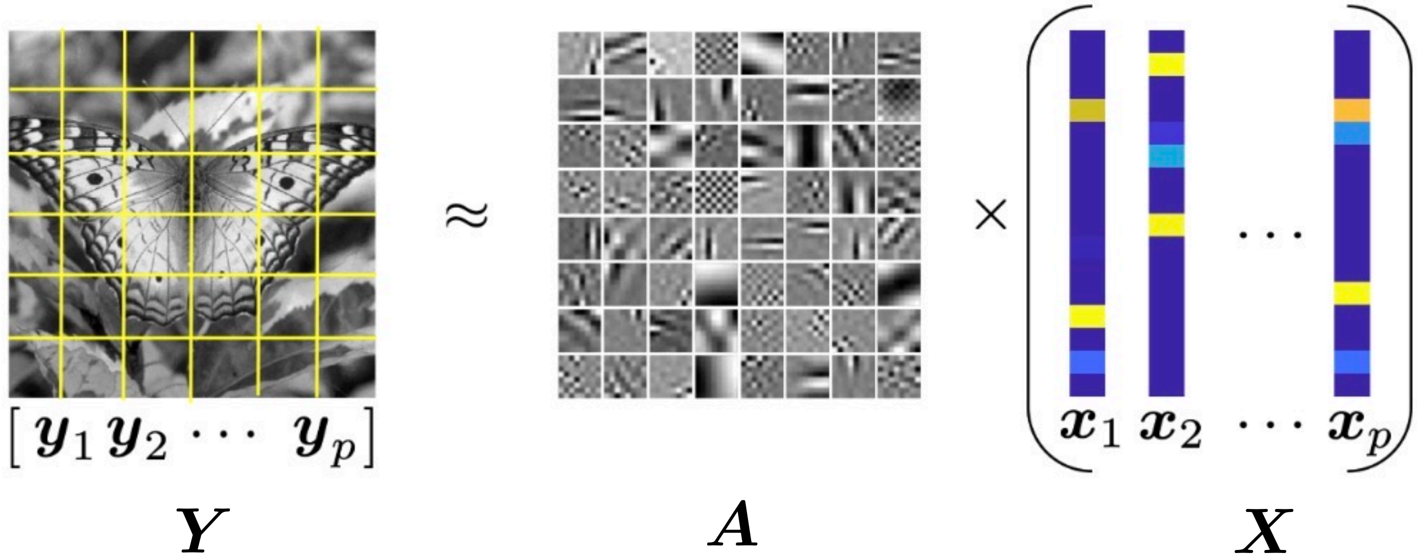
healthcare



audio representations

Data representation is ***critical*** for modern machine learning methods.

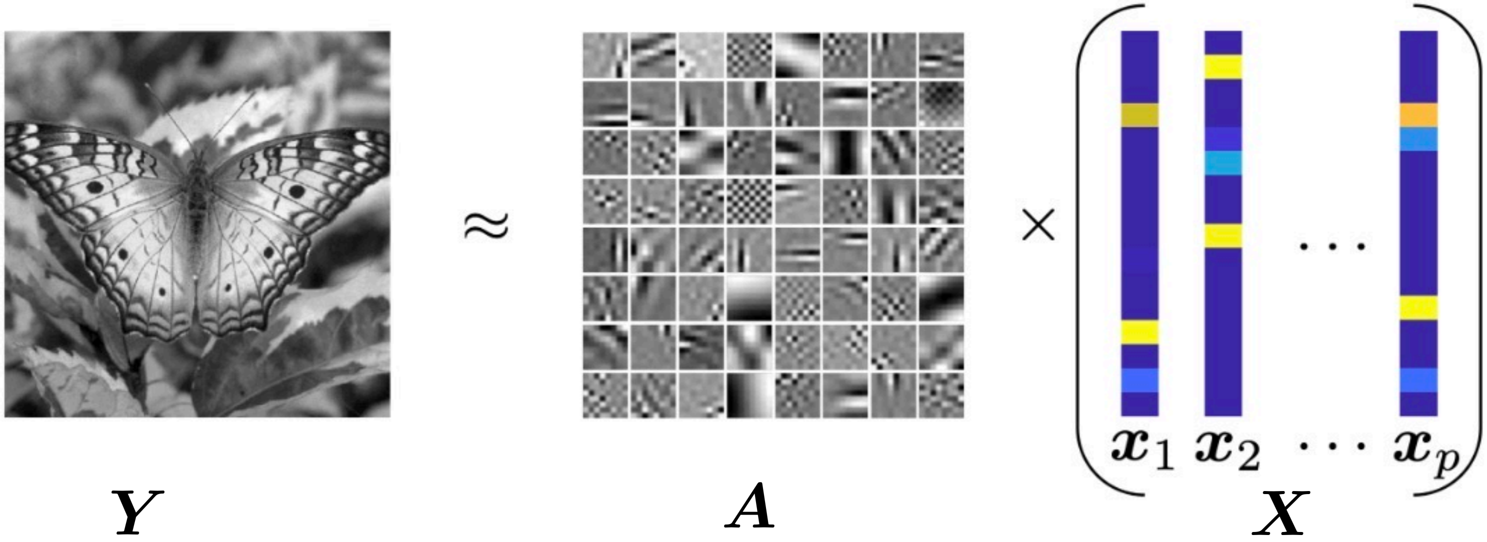
Unsupervised Learning



- **Learning sparsely-used dictionaries:**

  Given $\boldsymbol{Y} \in \mathbb{R}^{n\times p}$, jointly find overcomplete dictionary $\boldsymbol{A} \in \mathbb{R}^{n\times m}$ and sparse $\boldsymbol{X} \in \mathbb{R}^{m\times p}$.

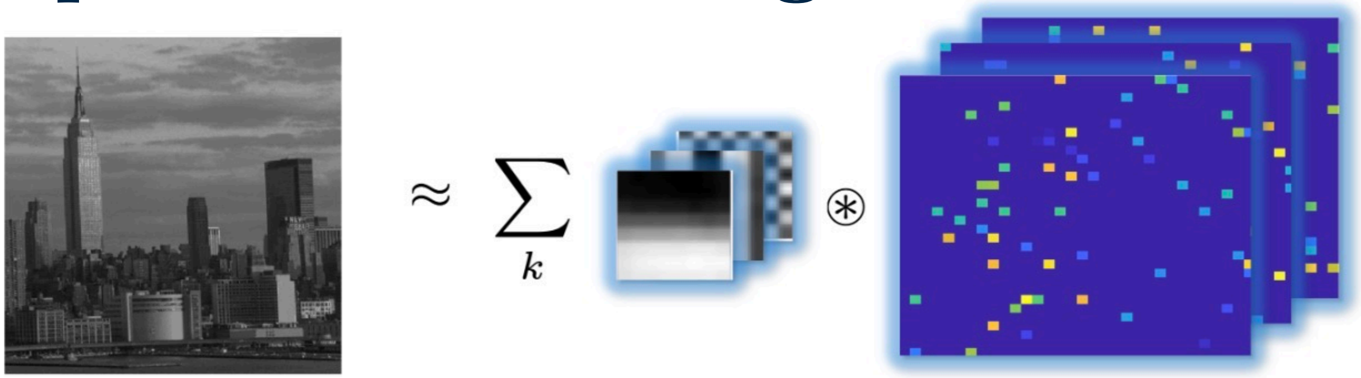
Unsupervised Learning



- **Learning sparsely-used dictionaries:**

$$\min_{\boldsymbol{A}\in\mathcal{M},\ \boldsymbol{X}} \underbrace{f(\boldsymbol{Y}, \boldsymbol{A}\cdot\boldsymbol{X})}_{\text{data fidelity}} + \lambda \cdot \underbrace{g(\boldsymbol{X})}_{\text{regularizer}}$$

Unsupervised Learning



$$\boldsymbol{y}_i \approx \sum_k \boldsymbol{a}_k \circledast \boldsymbol{x}_{ki}$$

- **Learning convolutional dictionaries:**
  Given $\{\boldsymbol{y}_i\}_i$, jointly learn convolutional dictionaries $\{\boldsymbol{a}_i\}_i$ and sparse coefficients $\{\boldsymbol{x}_{ki}\}_{i,k}$.

➢ Translation invariant, can be viewed as one layer of ConvNets

Denoising

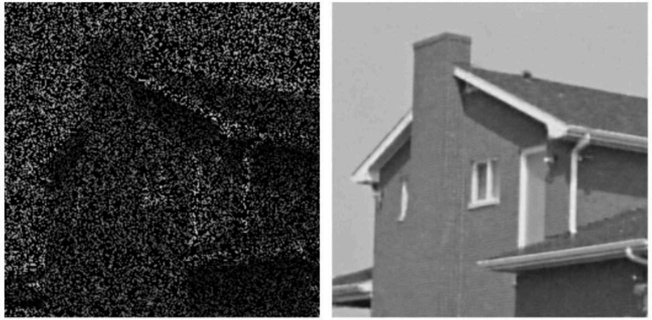


Image Restoration

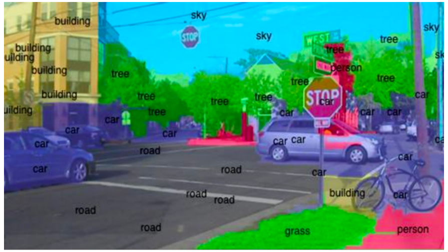
Super Resolution

Image Half-toning

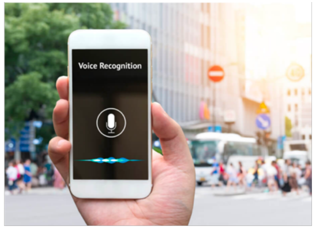
- Image courtesy of Julien Mairal et al.

Supervised (Deep) Learning

Deep learning has attained superior performances for many tasks in practice:



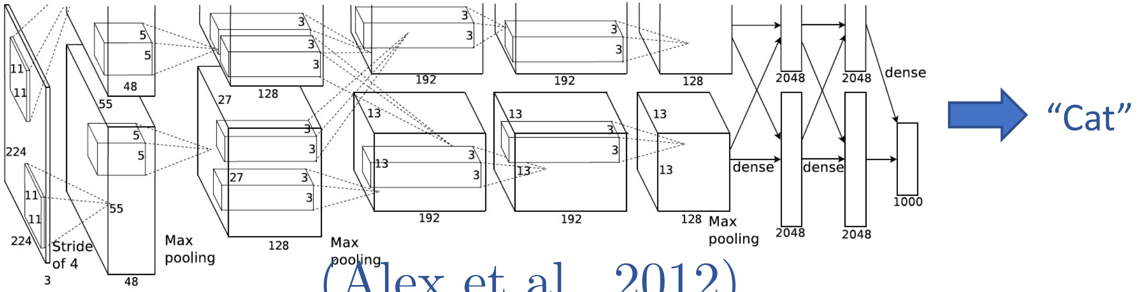
Computer vision



Natural language processing

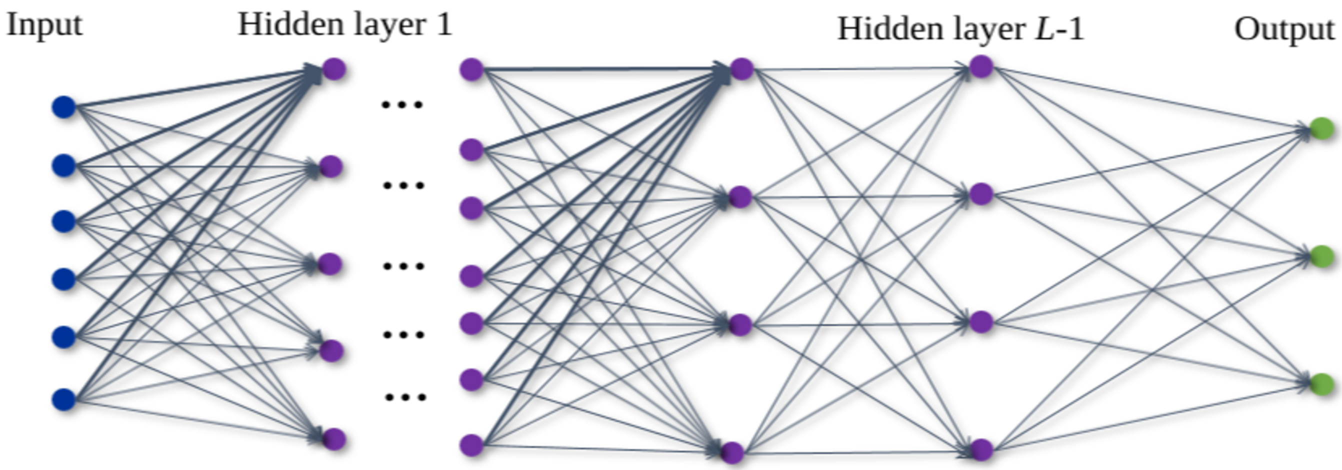
Gameplay

Protein modeling



(Alex et al., 2012)

Training Deep Neural Networks



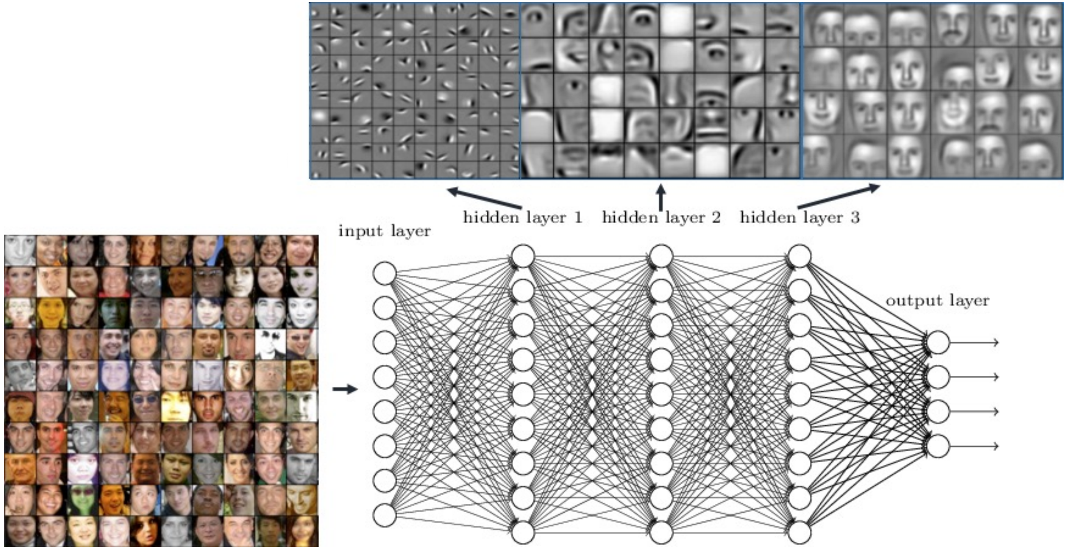
$$\psi_{\boldsymbol{\Theta}}(\boldsymbol{x}) \;=\; \boldsymbol{W}_L \sigma\left(\boldsymbol{W}_{L-1} \cdots \sigma(\boldsymbol{W}_1 \boldsymbol{x} + \boldsymbol{b}_1) + \boldsymbol{b}_{L-1}\right) + \boldsymbol{b}_L$$

$$\boldsymbol{\Theta} \;:=\; \{\boldsymbol{W}_\ell, \boldsymbol{b}_\ell\}_{\ell=1}^{L} \qquad \sigma(\cdot)\text{: nonlinear activations}$$

weights

bias

Training Deep Neural Networks



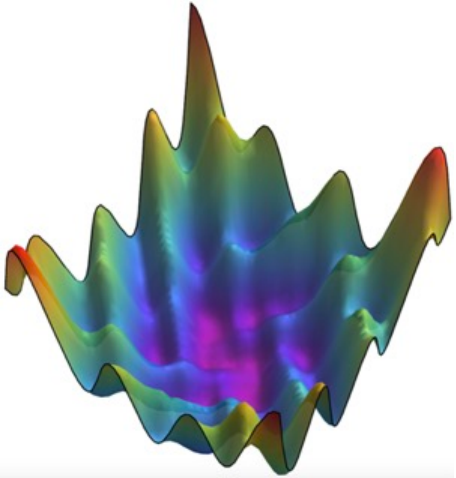
$$\min_{\boldsymbol{\Theta}} \sum_{k=1}^{K} \sum_{i=1}^{n_k} \mathcal{L}_{\mathrm{CE}}\left(\psi_{\boldsymbol{\Theta}}(\boldsymbol{x}_{k,i}), \boldsymbol{y}_k\right) + \lambda \left\|\boldsymbol{\Theta}\right\|_F^2$$

i-th input in the k-th class

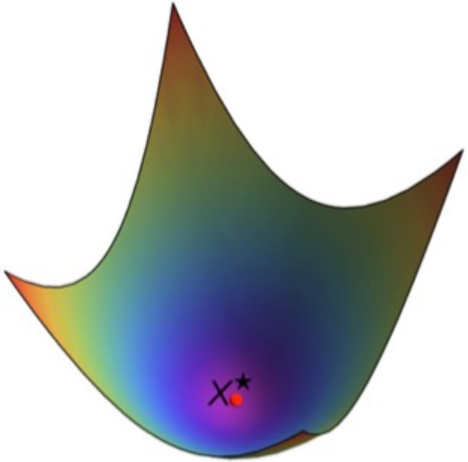
One-hot vector for the k-th class

Nonconvex Problems in Representation Learning

$$\boxed{\min_{\boldsymbol{x}} \; f(\boldsymbol{x}), \quad \text{s.t.} \;\; \boldsymbol{x} \in \mathbb{R}^n}$$

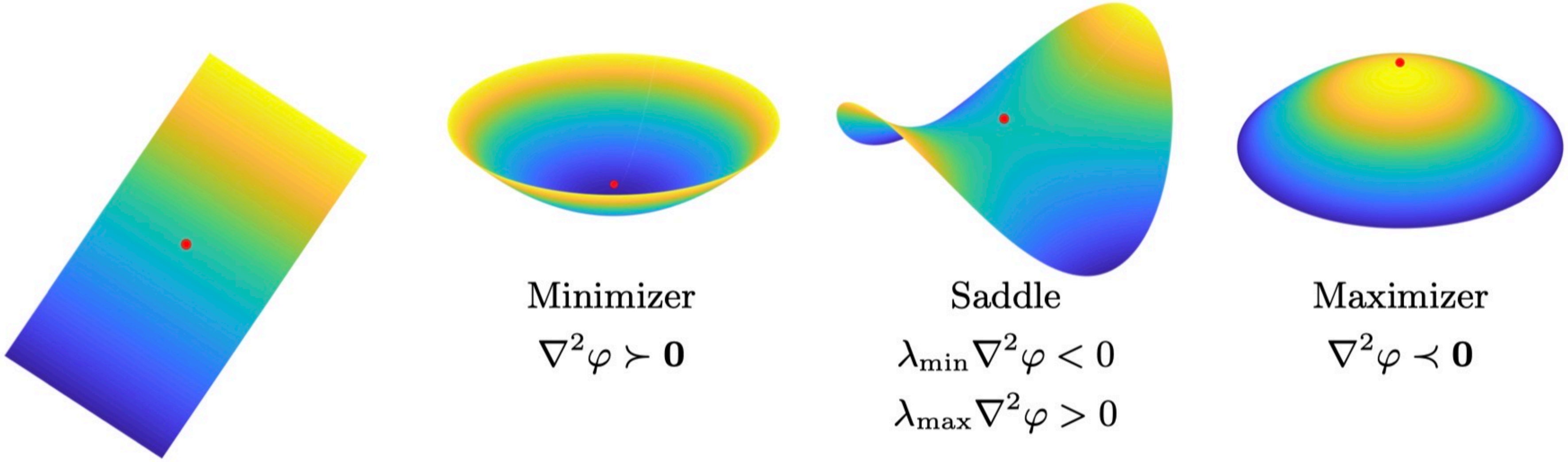


Nonconvex landscape



Convex landscape

General Nonconvex Problems



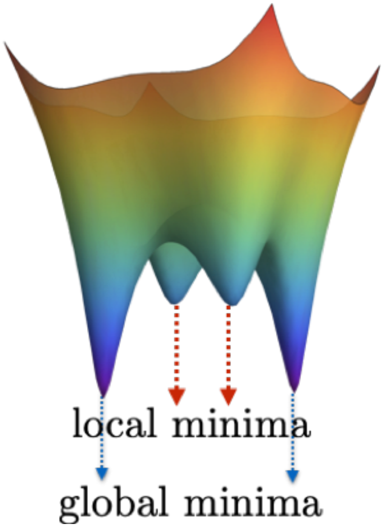
Noncritical Point ($\nabla \varphi \neq \mathbf{0}$)

Critical Points ($\nabla \varphi = \mathbf{0}$)

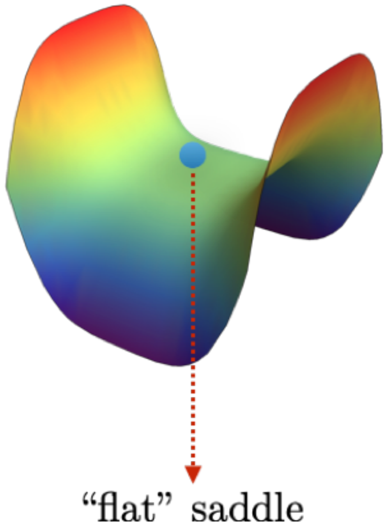
General Nonconvex Problems

$$\boxed{\min_{\boldsymbol{x}} \; f(\boldsymbol{x}), \quad \text{s.t.} \quad \boldsymbol{x} \in \mathbb{R}^n}$$

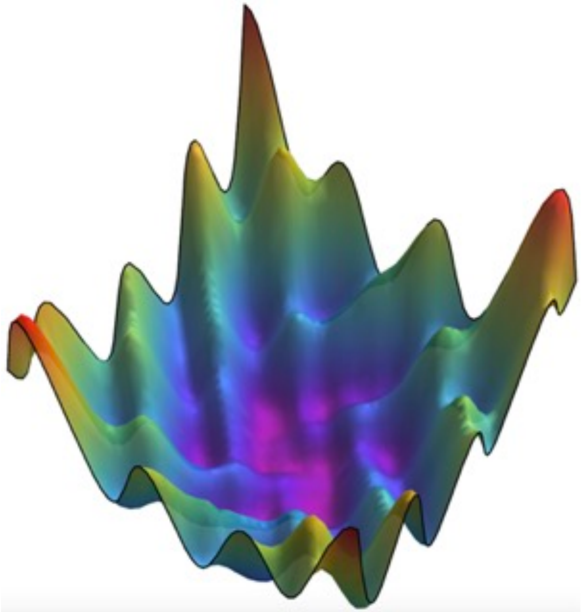
“bad” local minimizers



“flat” saddle points



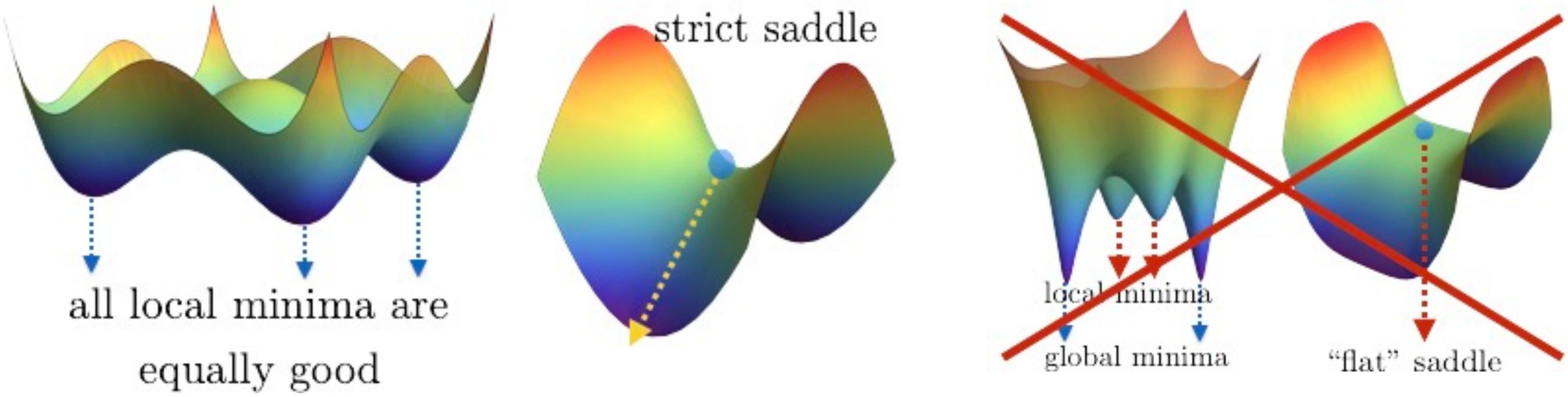
General Nonconvex Problems



$$\boxed{\min_{\boldsymbol{x}} \; f(\boldsymbol{x}), \quad \text{s.t.} \;\; \boldsymbol{x} \in \mathbb{R}^n}$$

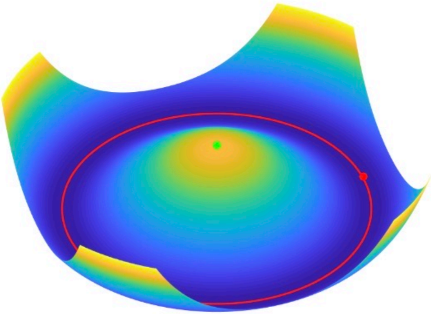
In the worst case, even finding a local minimizer is NP-hard (Murty et al. 1987)

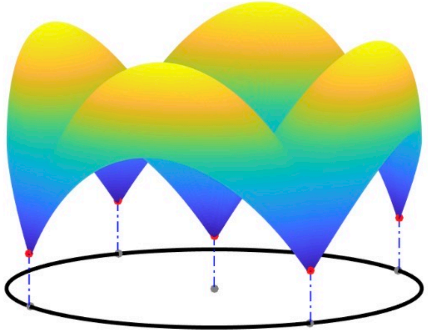
Optimizing Nonconvex Problems Globally



Benign nonconvex landscapes enable efficient global optimization!

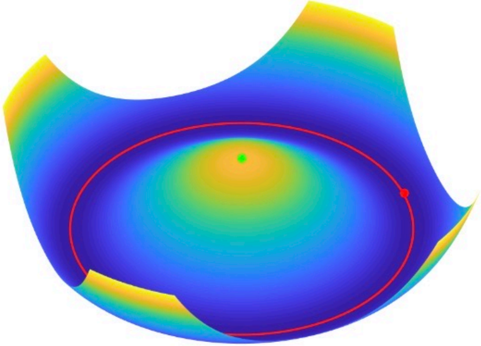
Nonconvex Problems with Benign Landscape

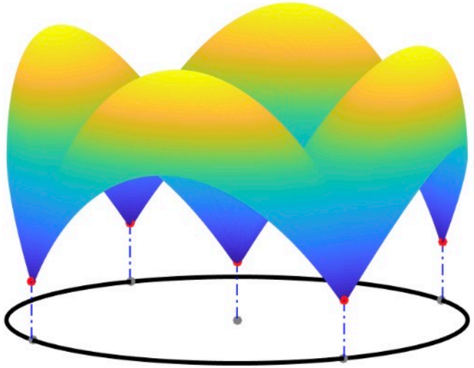




- Generalized Phase Retrieval [Sun'18]
- Low-rank Matrix Recovery [Ma'16, Jin'17, Chi'19]
- Sparse Dictionary Learning [Sun'16, Qu'20]
- (Orthogonal) Tensor Decomposition [Ge'15]
- Sparse Blind Deconvolution [Zhang'17, Li'18, Kuo'19]
- Deep Linear Network [Kawaguchi'16]
- …

Nonconvex Problems with Benign Landscape





- **Q. Qu** (*), Z. Zhu (*), X. Li, M. C. Tsakiris, J. Wright, R. Vidal, Finding the Sparsest Vectors in a Subspace: Theory, Algorithms, and Applications, In Submission, 2020.

- Y. Zhang, **Q. Qu**, J. Wright, Symmetry and Geometry in Nonconvex Optimization, In Submission, 2020.

Outline of this Talk

- **Learning Shallow Representations: (Convolutional) Dictionary Learning**

- Learning Deep Representations: Deep Neural Collapse

- Conclusion & Discussion

Landscape Analysis of Dictionary Learning

1. **Q. Qu**, Y. Zhai, X. Li, Y. Zhang, Z. Zhu, Analysis of optimization landscapes for overcomplete learning, *ICLR'20*, (oral, top 1.9%)
2. Y. Lau (*), **Q. Qu**(*), H. Kuo, P. Zhou, Y. Zhang, J. Wright, Short-and-sparse Deconvolution – A Geometric Approach, *ICLR'20*

- Provide the first **global nonconvex landscape** analysis for ***convolutional/overcomplete*** dictionary learning problems.

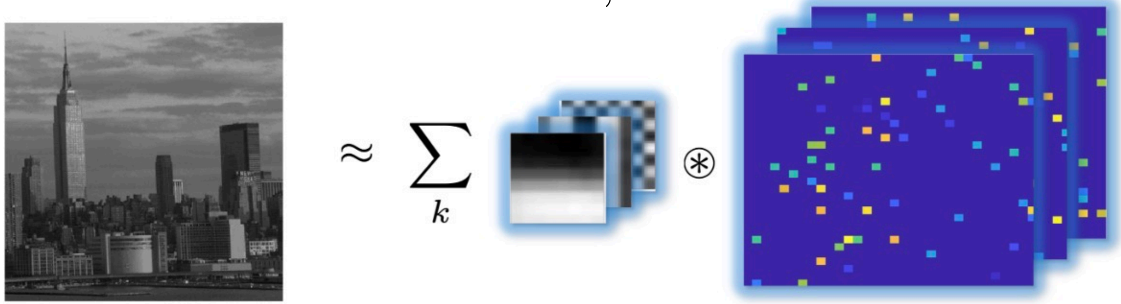
- Efficient nonconvex optimization methods to global solutions with applications in imaging.

Convolutional Dictionary Learning (DL)

Given multiple measurements $\{\boldsymbol{y}_i\}_i$ of circulant convolution

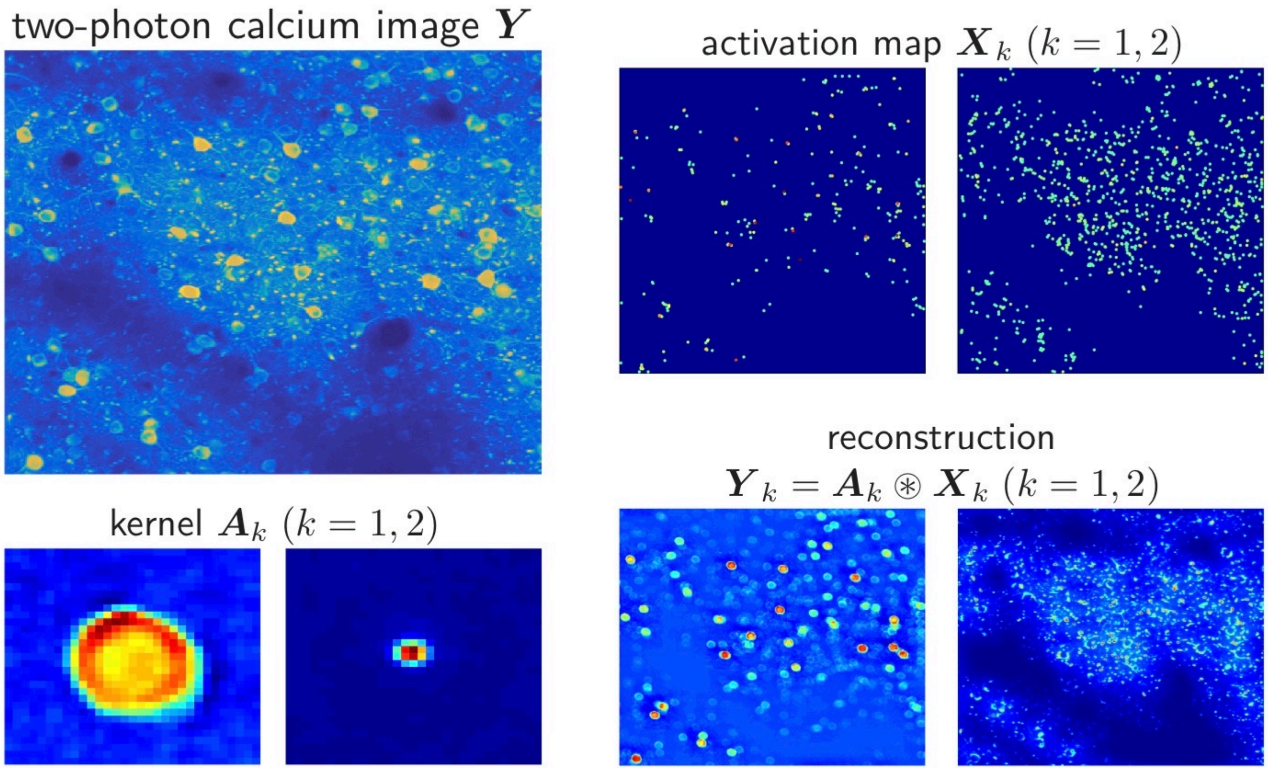
$$\boldsymbol{y}_i \quad = \quad \sum_{k=1}^{K} \boldsymbol{a}_k \circledast \boldsymbol{x}_{ki}, \qquad (1 \le i \le p)$$

can we learn all $\{\boldsymbol{a}_k\}_k$ and $\{\boldsymbol{x}_{ki}\}_{k,i}$ simultaneously?



$$\boldsymbol{y}_i \qquad \approx \sum_k \quad \boldsymbol{a}_k \quad \circledast \quad \boldsymbol{x}_{ki}$$

Convolutional Dictionary Learning (DL)



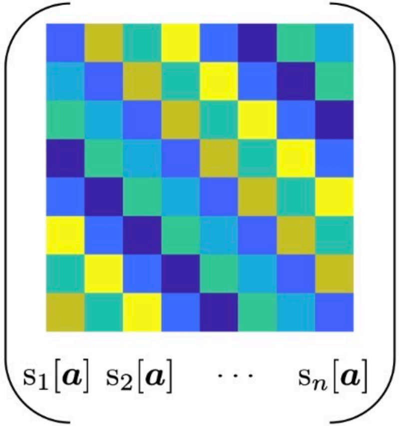
Convolutional DL vs. Overcomplete DL

For each $\boldsymbol{y}_i = \boldsymbol{a} \circledast \boldsymbol{x}_i$, we can *equivalently* rewrite in the matrix form as:

$$\boldsymbol{C}_{\boldsymbol{y}_i} \quad = \quad \boldsymbol{C}_{\boldsymbol{a}} \cdot \boldsymbol{C}_{\boldsymbol{x}_i}, \quad 1 \leq i \leq p$$

where a circulant matrix

$\boldsymbol{C}_{\boldsymbol{a}} =$



$\in \mathbb{R}^{n \times n}$

Convolutional DL vs. Overcomplete DL

For each sample

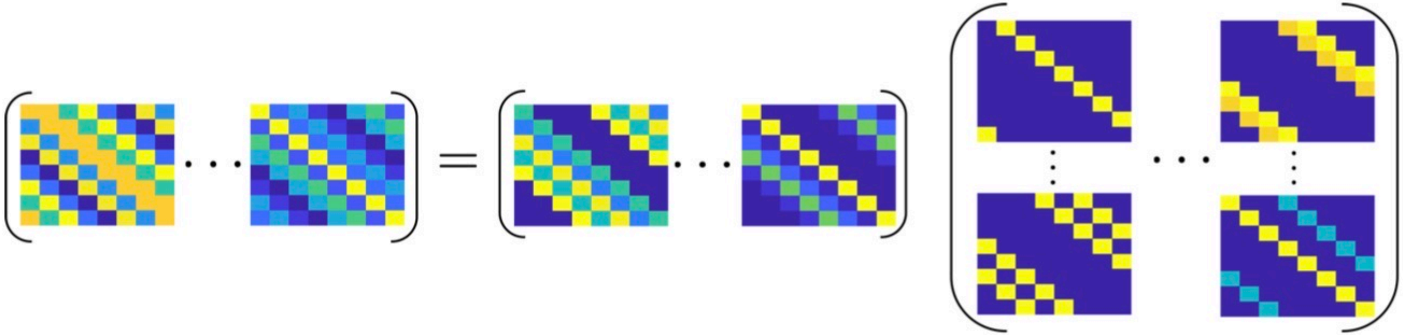
$$\boldsymbol{y}_i = \sum_{k=1}^{K} \boldsymbol{a}_k \circledast \boldsymbol{x}_{ki},$$

equivalently,

$$\boldsymbol{C}_{\boldsymbol{y}_i} = \underbrace{\begin{bmatrix} \boldsymbol{C}_{\boldsymbol{a}_1} & \cdots & \boldsymbol{C}_{\boldsymbol{a}_K} \end{bmatrix}}_{\textbf{overcomplete } \boldsymbol{A}_0} \cdot \underbrace{\begin{bmatrix} \boldsymbol{C}_{\boldsymbol{x}_{1i}} \\ \vdots \\ \boldsymbol{C}_{\boldsymbol{x}_{Ki}} \end{bmatrix}}_{\textbf{sparse } \boldsymbol{X}_i},$$

Convolutional DL vs. Overcomplete DL

Given $\boldsymbol{Y} = \boldsymbol{A}_0 \cdot \boldsymbol{X}_0$, learn overcomplete $\boldsymbol{A}_0$ and sparse $\boldsymbol{X}_0$?



$$\begin{bmatrix} \boldsymbol{C}_{\boldsymbol{y}_1} & \cdots & \boldsymbol{C}_{\boldsymbol{y}_p} \end{bmatrix} = \begin{bmatrix} \boldsymbol{C}_{\boldsymbol{a}_1} & \cdots & \boldsymbol{C}_{\boldsymbol{a}_K} \end{bmatrix} \begin{bmatrix} \boldsymbol{X}_1 & \cdots & \boldsymbol{X}_p \end{bmatrix}$$

data $\boldsymbol{Y}$ dictionary $\boldsymbol{A}_0$ sparse $\boldsymbol{X}_0$

Relationship to Dictionary Learning

We can find **one column** of $\boldsymbol{A}_0$ via

$$\boxed{\min_{\boldsymbol{q}} \; f_{DL}(\boldsymbol{q}) \;=\; -\left\|\boldsymbol{Y}^\top \boldsymbol{q}\right\|_4^4, \quad \text{s.t.} \quad \|\boldsymbol{q}\|_2 = 1.}$$

The underlying reasoning is that, in expectation

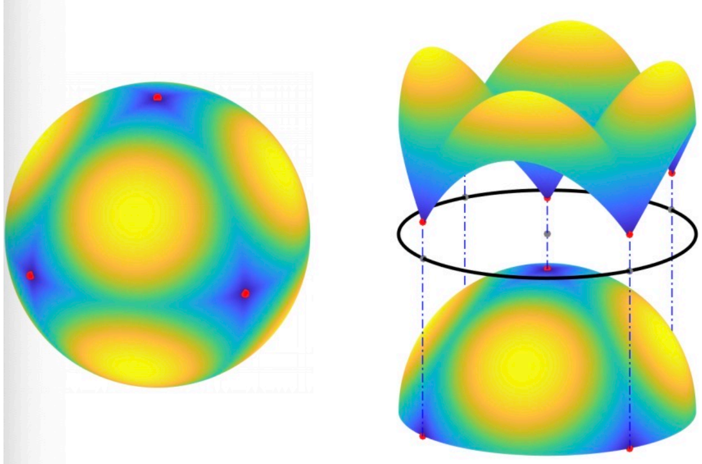
$$\mathbb{E}_{\boldsymbol{X}}\left[\left\|\boldsymbol{Y}^\top \boldsymbol{q}\right\|_4^4\right] \;=\; \mathbb{E}_{\boldsymbol{X}}\left[\left\|\boldsymbol{X}^\top \boldsymbol{A}_0^\top \boldsymbol{q}\right\|_4^4\right] \;=\; c_1 \left\|\boldsymbol{A}_0^\top \boldsymbol{q}\right\|_4^4 + c_2$$

for $\boldsymbol{X}$ following some sparse zero-mean distributions (e.g., Bernoulli-Gaussian)

Relationship to Dictionary Learning

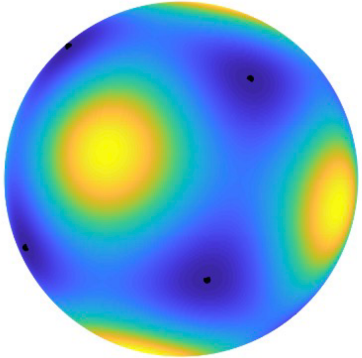
Given $\boldsymbol{A}_0 = \begin{bmatrix} \boldsymbol{a}_1 & \cdots & \boldsymbol{a}_m \end{bmatrix} \in \mathbb{R}^{n \times m}$

$$\boxed{\min_{\boldsymbol{q}} \; - \left\| \boldsymbol{A}_0^\top \boldsymbol{q} \right\|_4^4, \quad \text{s.t.} \quad \left\| \boldsymbol{q} \right\|_2 = 1.}$$



- When $m \leq n$, and $\{\boldsymbol{a}_i\}_{i=1}^m$ are **orthogonal**, existing result [Ge'15] has shown that the function is a **strict saddle function** with benign optimization landscape, all global solutions are approximately $\{\pm \boldsymbol{a}_i\}_{i=1}^m$.
- The analysis of orthogonal case **cannot** be generalized to overcomplete settings.

Global Landscape of Overcomplete DL



$$\boxed{\min_{\boldsymbol{q}} \; f_{DL}(\boldsymbol{q}) \;=\; -\left\|\boldsymbol{Y}^\top \boldsymbol{q}\right\|_4^4, \quad \text{s.t.} \quad \|\boldsymbol{q}\|_2 = 1.}$$

Theorem (Informal) *Suppose that (i) $K = m/n$ is a constant, (ii) $\boldsymbol{A}_0$ is near orthogonal, and (iii) $p \geq \Omega(\mathrm{poly}(n))$. Then with high probability every critical point of $f(\boldsymbol{q})$ is either*

- *a **strict saddle point** exhibits negative curvature;*
- *or close to a **target solution**: one column $\boldsymbol{a}_i$ of $\boldsymbol{A}$.*

Assumptions on $\boldsymbol{A}$ (Near Orthogonal)

- Row orthogonal: unit norm **tight frame** (UNTF)

$$\sqrt{\frac{n}{m}}\boldsymbol{A}_0\boldsymbol{A}_0^\top = \boldsymbol{I}, \quad \|\boldsymbol{a}_i\|_2 = 1.$$

- **Incoherence** of the columns (near orthogonal)

$$\max_{i\neq j}|\langle \boldsymbol{a}_i, \boldsymbol{a}_j\rangle| \le \mu.$$

Overcomplete Dictionary Learning

Given $\boldsymbol{Y} = \boldsymbol{A}_0 \cdot \boldsymbol{X}_0$, learn overcomplete $\boldsymbol{A}_0$ and sparse $\boldsymbol{X}_0$?

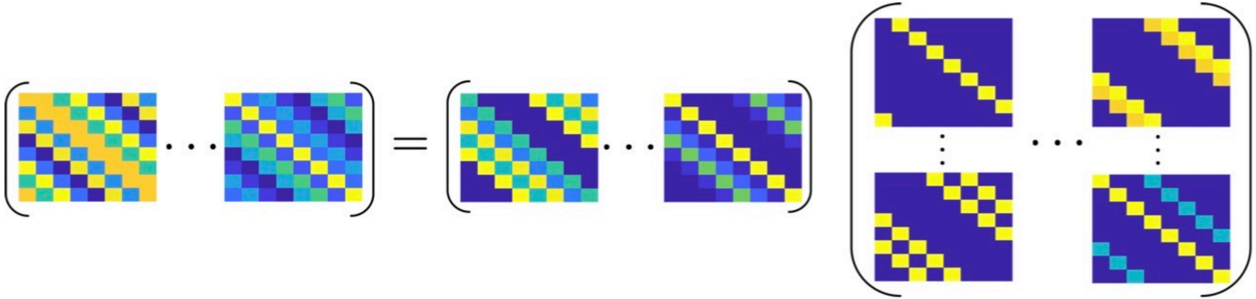
$$\boldsymbol{A}_0 \in \mathbb{R}^{n \times m}$$

	Complete $n = m$	Overcomplete $m > n$
Global	[Sun, **Qu**, Wright'16] [Sun, **Qu**, Wright'16] [Li et al.'18]	**Our Result**
Local	[**Qu**, Sun, Wright'16] [Zhai et al.'19]	**Initialization Required** [Arora et al.'14&15] [Agarwal et al.'16] [Chatterji et al.'17] [Awasthi et al.'18]

From Overcomplete DL to Convolutional DL

Find one shift of the kernel $\boldsymbol{a}_i$ via

$$\boxed{\min_{\boldsymbol{q}} \; f_{CDL}(\boldsymbol{q}) \; = \; -\left\|\boldsymbol{q}^\top \boldsymbol{Y}\right\|_4^4, \quad \text{s.t.} \quad \boldsymbol{q} \in \mathbb{S}^{n-1}}$$



$$\begin{pmatrix} \boldsymbol{C}_{\boldsymbol{y}_1} & \cdots & \boldsymbol{C}_{\boldsymbol{y}_p} \end{pmatrix} = \begin{pmatrix} \boldsymbol{C}_{\boldsymbol{a}_1} & \cdots & \boldsymbol{C}_{\boldsymbol{a}_K} \end{pmatrix} \begin{pmatrix} \boldsymbol{X}_1 & \cdots & \boldsymbol{X}_p \end{pmatrix}$$

data $\boldsymbol{Y}$ **dictionary $\boldsymbol{A}_0$** **sparse $\boldsymbol{X}_0$**

Convolutional Dictionary Learning

Find one shift of the kernel $\boldsymbol{a}_i$ via

$$\boxed{\min_{\boldsymbol{q}} \; f_{CDL}(\boldsymbol{q}) \;=\; -\left\|\boldsymbol{q}^\top \boldsymbol{P}\boldsymbol{Y}\right\|_4^4, \quad \text{s.t.} \quad \boldsymbol{q} \in \mathbb{S}^{n-1}}$$

- Preconditioning matrix:

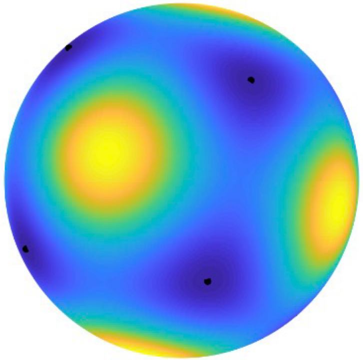
$$\boldsymbol{P} \;=\; \left((\theta n K)^{-1}\boldsymbol{Y}\boldsymbol{Y}^\top\right)^{-1/2} \;\approx\; \left(\boldsymbol{A}_0\boldsymbol{A}_0^\top\right)^{-1/2}$$

- Effective dictionary is tight frame (but not necessarily unit norm)

$$\boldsymbol{P}\boldsymbol{Y} \;\approx\; \underbrace{\left(\boldsymbol{A}_0\boldsymbol{A}_0^\top\right)^{-1/2}\boldsymbol{A}_0}_{\boldsymbol{A}} \; \boldsymbol{X}_0 \;=\; \boldsymbol{A}\,\boldsymbol{X}_0$$

Convolutional Dictionary Learning

Find one shift of the kernel $\boldsymbol{a}_i$ via



$$\boxed{\min_{\boldsymbol{q}} \; -\left\| \boldsymbol{q}^\top \boldsymbol{A} \boldsymbol{X}_0 \right\|_4^4, \quad \text{s.t.} \quad \boldsymbol{q} \in \mathbb{S}^{n-1}}$$

- Preconditioning matrix:

$$\boldsymbol{P} = \left((\theta n K)^{-1} \boldsymbol{Y}\boldsymbol{Y}^\top \right)^{-1/2} \approx \left(\boldsymbol{A}_0 \boldsymbol{A}_0^\top \right)^{-1/2}$$

- Effective dictionary is tight frame (but not necessarily unit norm)

$$\boldsymbol{P}\boldsymbol{Y} \approx \underbrace{\left(\boldsymbol{A}_0 \boldsymbol{A}_0^\top \right)^{-1/2} \boldsymbol{A}_0}_{\boldsymbol{A}} \boldsymbol{X}_0 = \boldsymbol{A} \, \boldsymbol{X}_0$$

Local Landscape of Convolutional DL

Theorem (Informal) *Suppose that (i) $K = m/n$ is a constant, (ii) $\boldsymbol{A}$ is near orthogonal, and (iii) $p \geq \Omega(\mathrm{poly}(n))$.* **Locally**, *every critical point of $f_{CDL}(\boldsymbol{q})$ is either*

- *a **strict saddle point** exhibits negative curvature;*
- *or close to a **target solution**: a precond. shift of $\boldsymbol{a}_i$.*

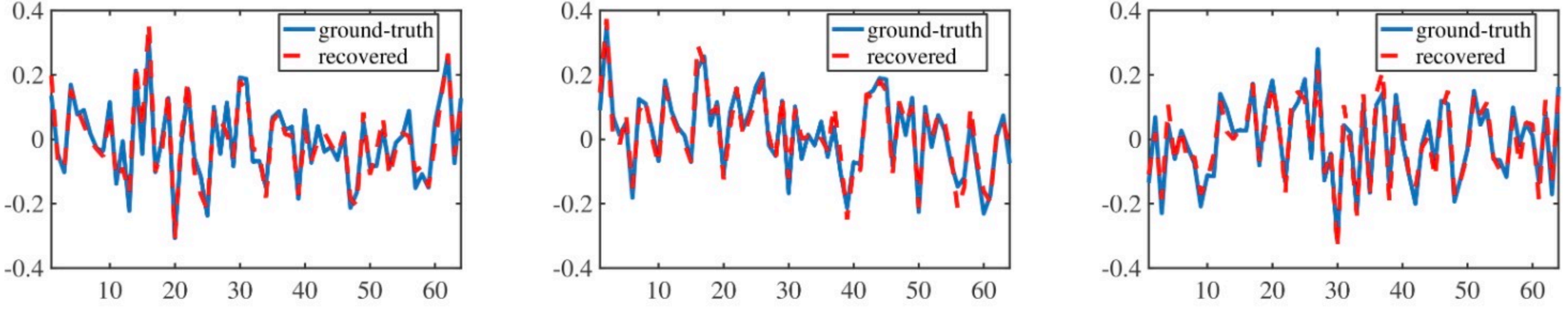
- We show the result over a local level-set

$$\mathcal{R}_{\mathrm{CDL}} := \left\{ \boldsymbol{q} \in \mathbb{S}^{n-1} \mid \mathbb{E}_{\boldsymbol{X}} \left[f_{CDL}(\boldsymbol{q}) \right] \leq -c \left\| \boldsymbol{A}^\top \boldsymbol{q} \right\|_3^2 \right\},$$

- We can cook up smart yet simple initializations, with all future iterations stay in the region.

Learning Random Filters

$$\boxed{\min_{\boldsymbol{q}} \; f_{CDL}(\boldsymbol{q}) \;=\; -\left\|\boldsymbol{q}^\top \boldsymbol{P}\boldsymbol{Y}\right\|_4^4, \quad \text{s.t.} \quad \boldsymbol{q} \in \mathbb{S}^{n-1}}$$



Filter 1 Filter 2 Filter 3

Learning 3 random filters by the proposed approach.

From Theory to Practical Methods

- Recovering one filter:

$$\boxed{\min_{\boldsymbol{q}} \; f_{CDL}(\boldsymbol{q}) \; = \; -\left\|\boldsymbol{q}^\top \boldsymbol{P}\boldsymbol{Y}\right\|_4^4, \quad \text{s.t.} \quad \boldsymbol{q} \in \mathbb{S}^{n-1}}$$

- Finding all filters via Bilinear Lasso formulation:

$$\boxed{\min_{\boldsymbol{a}_k, \boldsymbol{x}_k} \frac{1}{2}\left\|\boldsymbol{y} - \sum_{k=1}^{N} \boldsymbol{a}_k \circledast \boldsymbol{x}_k\right\|^2 + \lambda \sum_{k=1}^{N} \left\|\boldsymbol{x}_k\right\|_1, \quad \text{s.t.} \; \left\|\boldsymbol{a}_k\right\| = 1.}$$

From Theory to Practical Methods

- Finding all filters via Bilinear Lasso formulation:

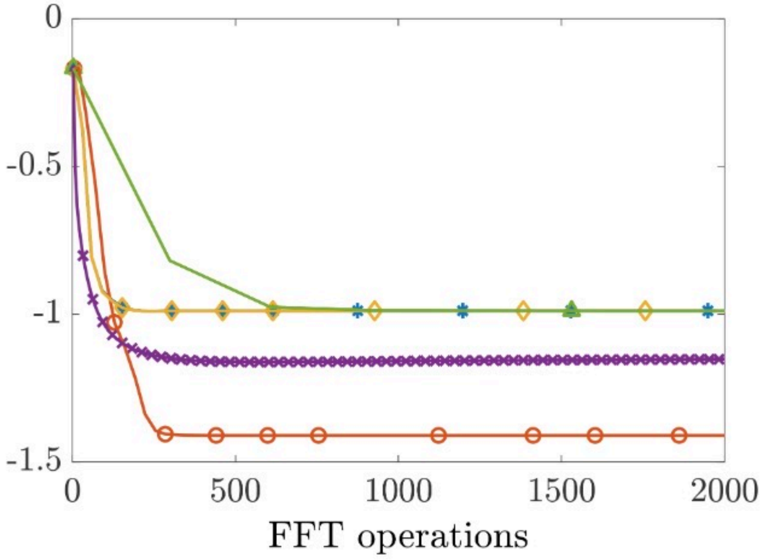
$$\boxed{\min_{\boldsymbol{a}_k, \boldsymbol{x}_k} \frac{1}{2} \left\| \boldsymbol{y} - \sum_{k=1}^{N} \boldsymbol{a}_k \circledast \boldsymbol{x}_k \right\|^2 + \lambda \sum_{k=1}^{N} \left\| \boldsymbol{x}_k \right\|_1, \quad \text{s.t. } \left\| \boldsymbol{a}_k \right\| = 1.}$$

- Optimization. Alternating descent method

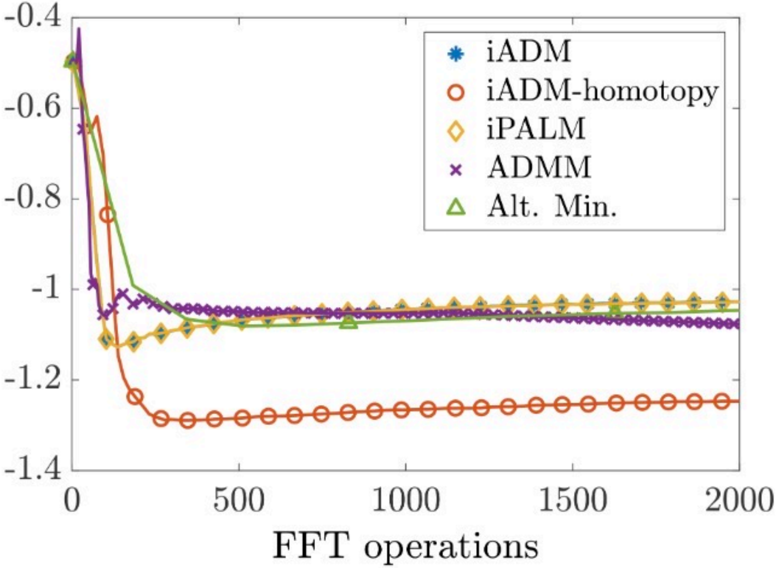
$$\begin{aligned} \boldsymbol{x} &\leftarrow \operatorname{prox}\left(\boldsymbol{x} - \tau \cdot \nabla_{\boldsymbol{x}} \varphi_{\mathrm{BL}}(\boldsymbol{a}, \boldsymbol{x})\right) \\ \boldsymbol{a} &\leftarrow \mathcal{P}_{\mathbb{S}^{n-1}}\left(\boldsymbol{a} - t \cdot \operatorname{grad}_{\boldsymbol{a}} \varphi_{\mathrm{BL}}(\boldsymbol{a}, \boldsymbol{x})\right), \end{aligned}$$

with few extra caveats.

From Theory to Practical Methods



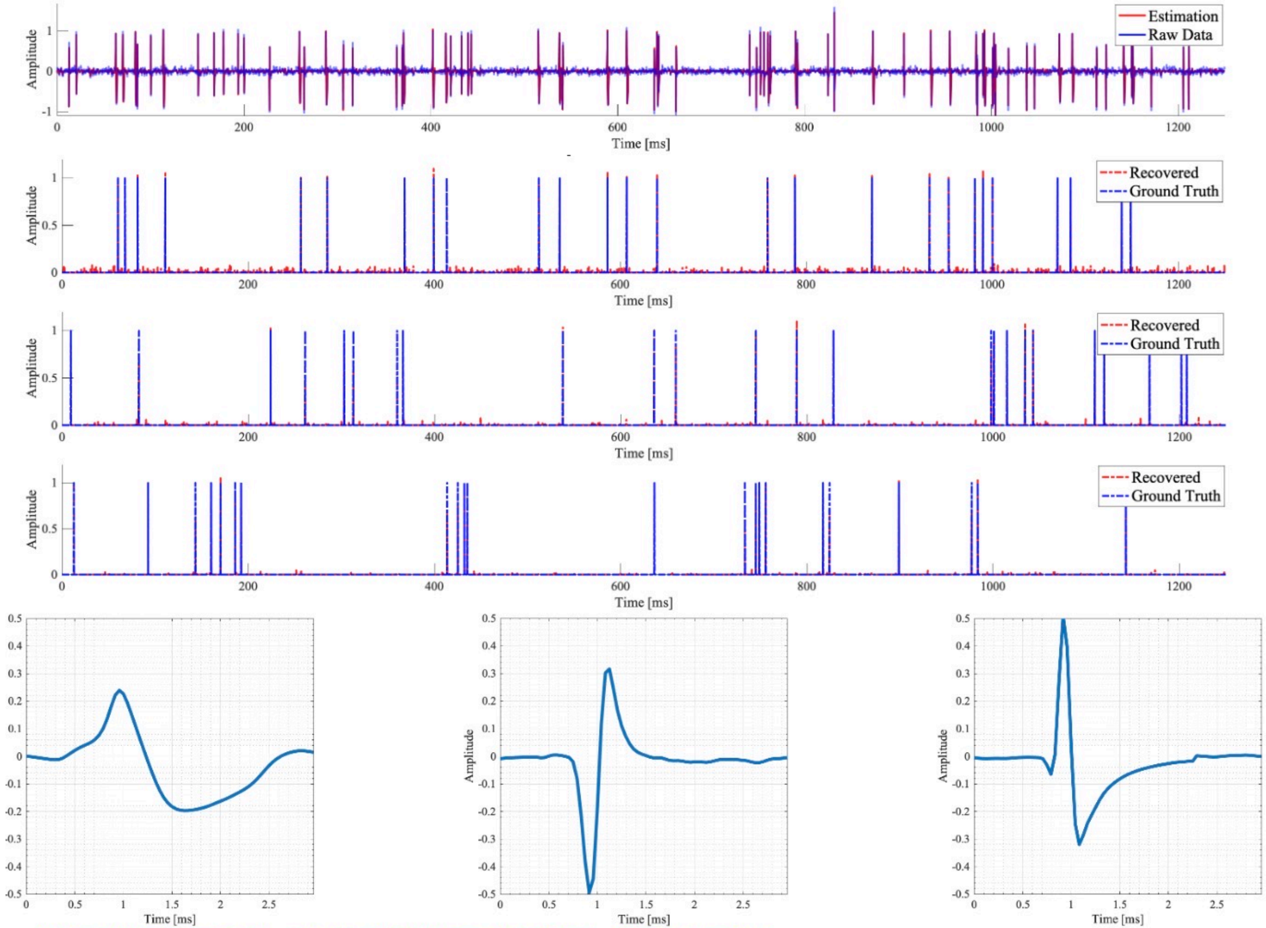
all-pass filter



low-pass filter

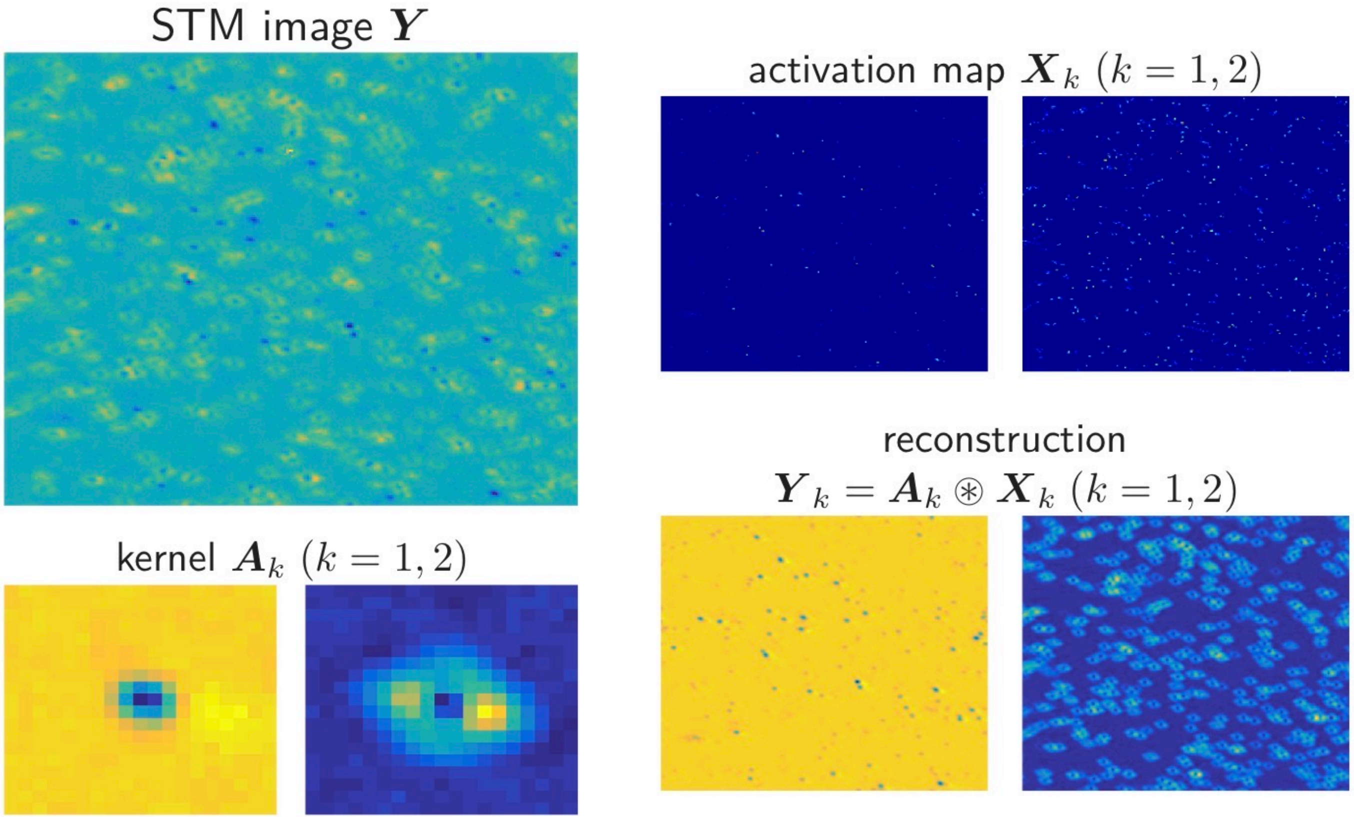
Comparison of convergence (time).

Spike Sorting



- https://vis.caltech.edu/~rodri/Wave_clus/Wave_clus_home.htm

Defects Detection in Scan Tunneling Microscopy



Outline of this Talk

- Learning Shallow Representations: (Convolutional) Dictionary Learning
- **Learning Deep Representations: Deep Neural Collapse**
- Conclusion & Discussion

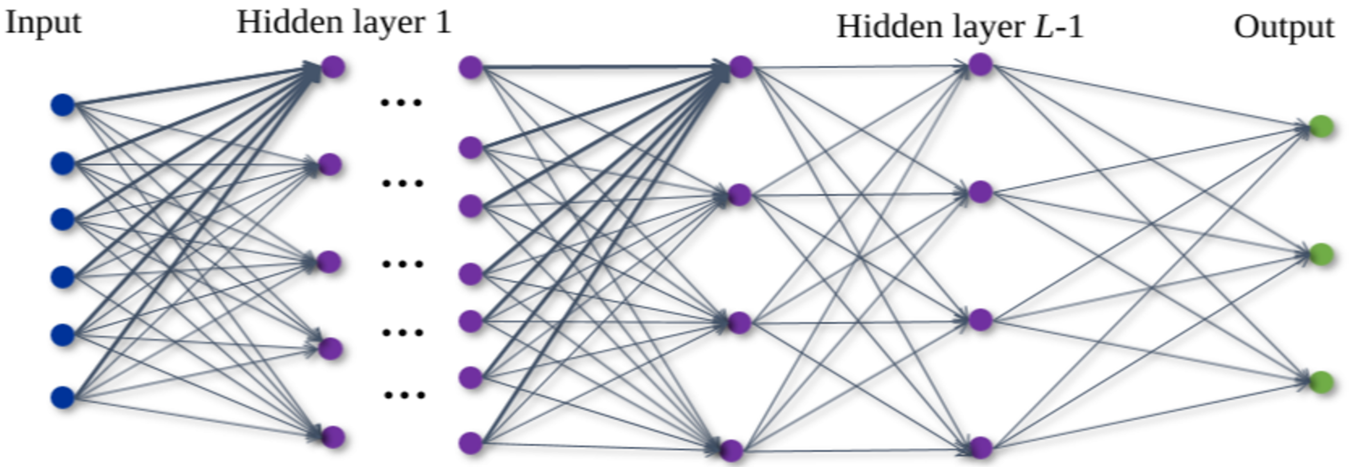
Understanding Deep Neural Networks

Z. Zhu, T. Ding, J. Zhou, X. Li, C. You, J. Sulam, and **Q. Qu**, A Geometric Analysis of Neural Collapse with Unconstrained Features, *arXiv Preprint arXiv:2105.02375*, May 2021.

- Analyzes the **global landscape** of the training loss based on the **unconstrained feature model**

- Explains the ubiquity of **Neural Collapse** of the learned representations of the network

Understanding Deep Neural Networks



$$\psi_{\boldsymbol{\Theta}}(\boldsymbol{x}) \;=\; \boldsymbol{W}_L \sigma\left(\boldsymbol{W}_{L-1}\cdots\sigma(\boldsymbol{W}_1\boldsymbol{x}+\boldsymbol{b}_1)+\boldsymbol{b}_{L-1}\right)+\boldsymbol{b}_L$$

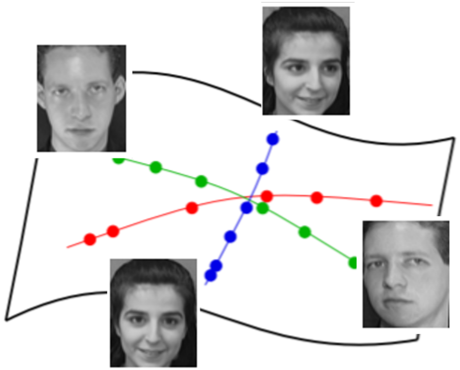
$$\boldsymbol{\Theta} \;:=\; \{\boldsymbol{W}_\ell, \boldsymbol{b}_\ell\}_{\ell=1}^{L} \qquad \sigma(\cdot)\text{: nonlinear activations}$$

weights

bias

Terminology for Classification

- Labels: $k = 1, \ldots, K$
  - ➢ $K = 10$ classes (MNIST, CIFAR10, etc.)
  - ➢ $K = 1000$ classes (ImageNet)



Data in the input space

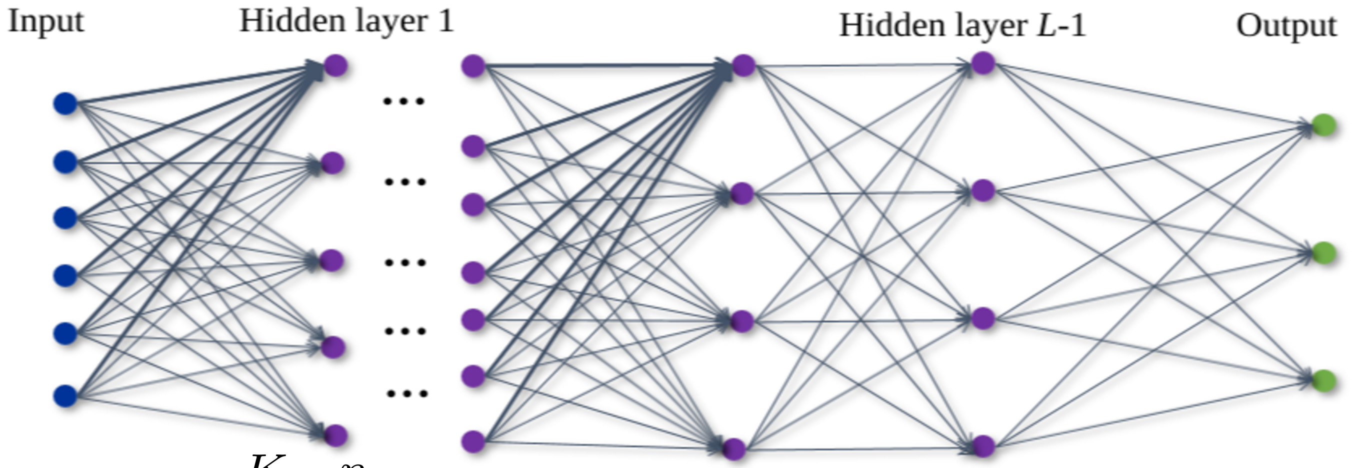
$$\boldsymbol{x} \xrightarrow[\psi_{\boldsymbol{\Theta}}(\boldsymbol{x})]{} \boldsymbol{y}$$

Neural network

$$\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} \cdots \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

One-hot vectors in $\mathbb{R}^K$

Understanding Deep Neural Networks



$$\min_{\boldsymbol{\Theta}} \sum_{k=1}^{K} \sum_{i=1}^{n_k} \mathcal{L}_{\mathrm{CE}}\left(\psi_{\boldsymbol{\Theta}}(\boldsymbol{x}_{k,i}), \boldsymbol{y}_k\right) + \lambda \|\boldsymbol{\Theta}\|_F^2$$

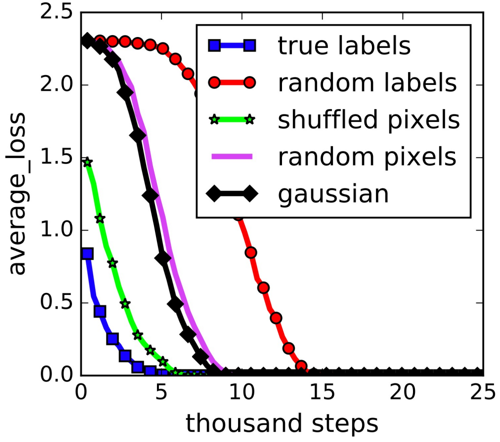
i-th input in the k-th class

One-hot vector for the k-th class

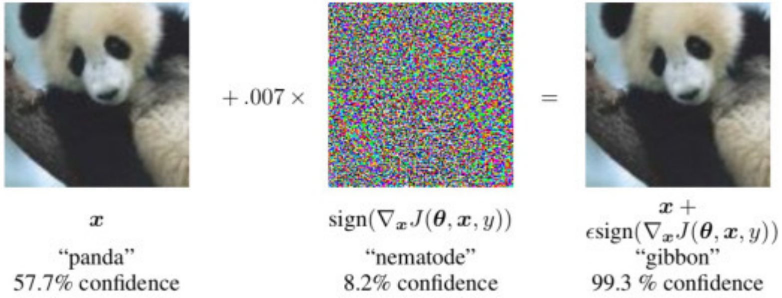
Mysteries in Deep Learning

- **Architecture design** (before training):
  - ➢ Feature dimensionality
  - ➢ Network width and depth
  - ➢ Activation functions
- **Optimization** (during training):
  - ➢ Choices of loss functions
  - ➢ Optimization algorithms, normalization
- **Properties of learned network**

(after training):
  - ➢ Generalization
  - ➢ Robustness

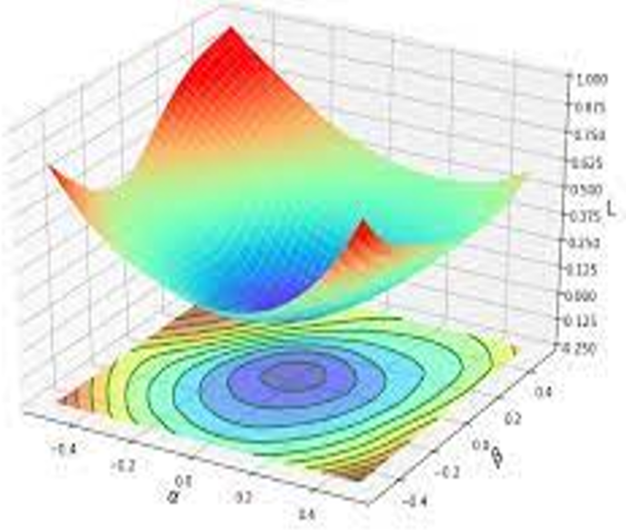


[Zhang et al ICLR'17]

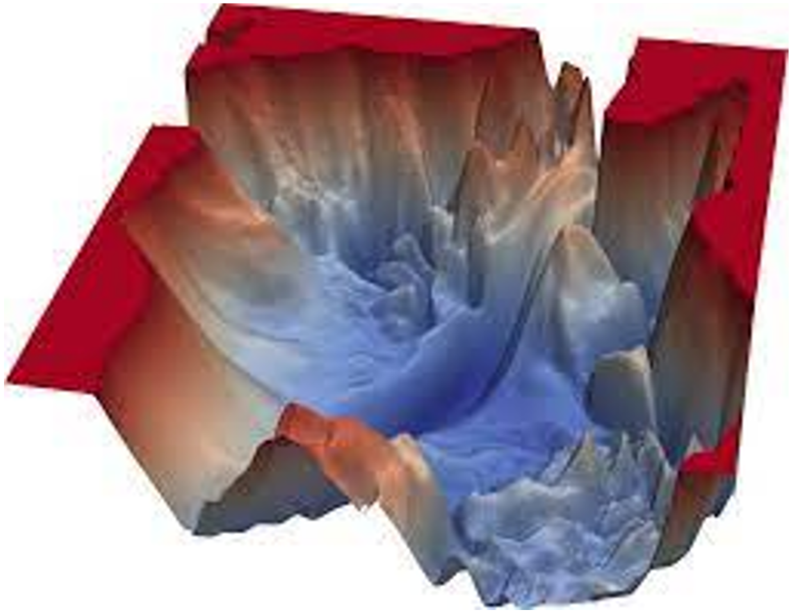


Goodfellow et al ICLR'15

Fundamental Challenges: Optimization



Landscape in Classical Optimization
(abundant algorithms & theory)



Landscape of Modern Deep Neural Networks
Credited to [Li'17]

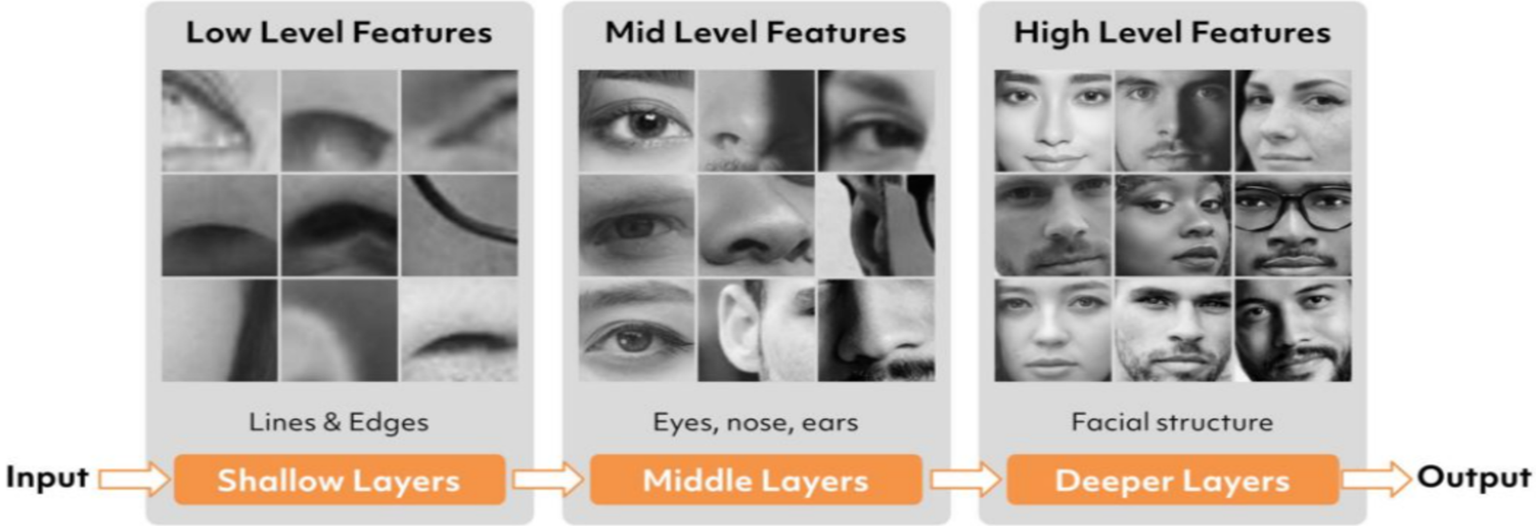
Optimization: Existing Results

Existing analysis are based on various **simplifications**:

- **Go Linear:** deep linear networks [Kawaguchi'16], deep matrix factorizations [Arora'19], etc.
- **Go Shallow:** Two-layer neural networks [Safran'18, Liang'18], etc.
- **Go Wide:** Neural tangent kernels [Jacot'18, Allen-Zhu'18, Du'19], mean-field analysis [Mei'19, Sirignano'19], etc.

Most of results ***hardly*** provide much insights for **practical** neural networks.

Features – What NNs (Conceptually) Designed to Learn



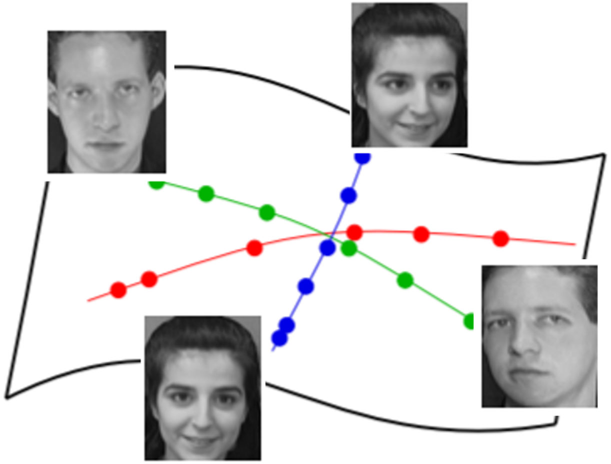
Wishful Design: NNs learn rich feature representations across different levels?

Neural Collapse in Classification

$$\psi_{\boldsymbol{\Theta}}(\boldsymbol{x}) = \boldsymbol{W}_L \underbrace{\sigma\left(\boldsymbol{W}_{L-1} \cdots \sigma(\boldsymbol{W}_1 \boldsymbol{x} + \boldsymbol{b}_1) + \boldsymbol{b}_{L-1}\right)}_{\phi_{\boldsymbol{\theta}}(\boldsymbol{x}) =: \boldsymbol{h}} + \boldsymbol{b}_L$$

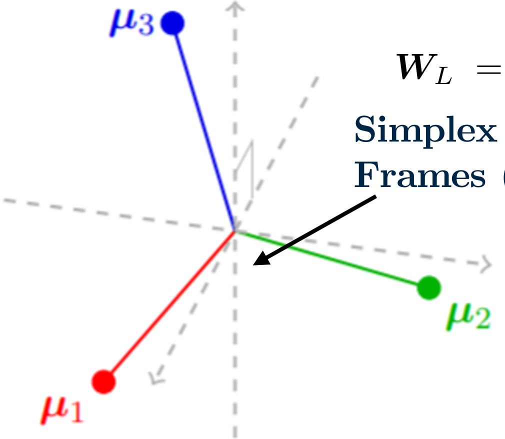
Last-layer classifier

Last-layer feature



Data in the Input Space

$$\xrightarrow{}$$
$\phi_{\boldsymbol{\theta}}(\cdot)$



$$\boldsymbol{W}_L = \begin{bmatrix} \boldsymbol{\mu}_1 & \cdots & \boldsymbol{\mu}_K \end{bmatrix}^\top.$$

Simplex Equiangular Tight Frames (Simplex ETF)

Neural Collapse
in the Feature Space

Neural Collapse in Classification

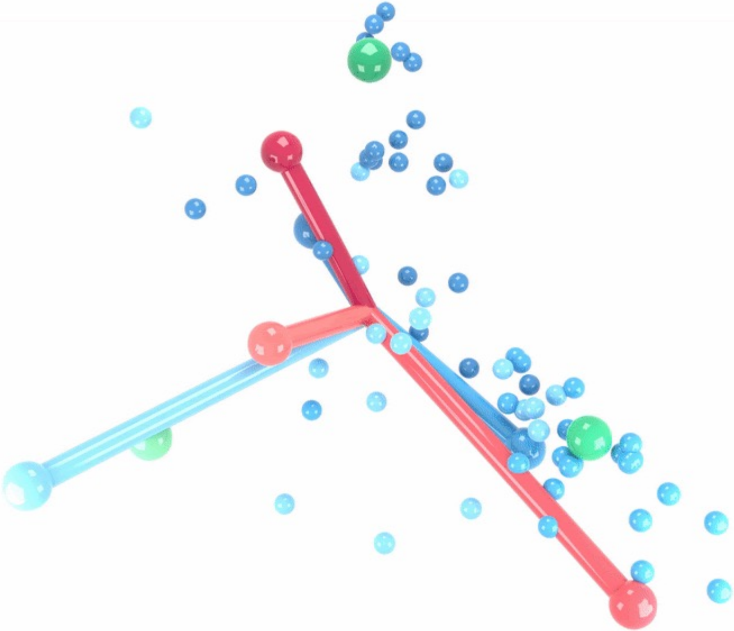
Prevalence of neural collapse during the terminal phase of deep learning training

Vardan Papyan, X. Y. Han, and David L. Donoho

+ See all authors and affiliations

PNAS October 6, 2020 117 (40) 24652-24663; first published September 21, 2020;
https://doi.org/10.1073/pnas.2015509117

Contributed by David L. Donoho, August 18, 2020 (sent for review July 22, 2020; reviewed by Helmut Boelsckei and Stéphane Mallat)



Neural Collapse in Classification

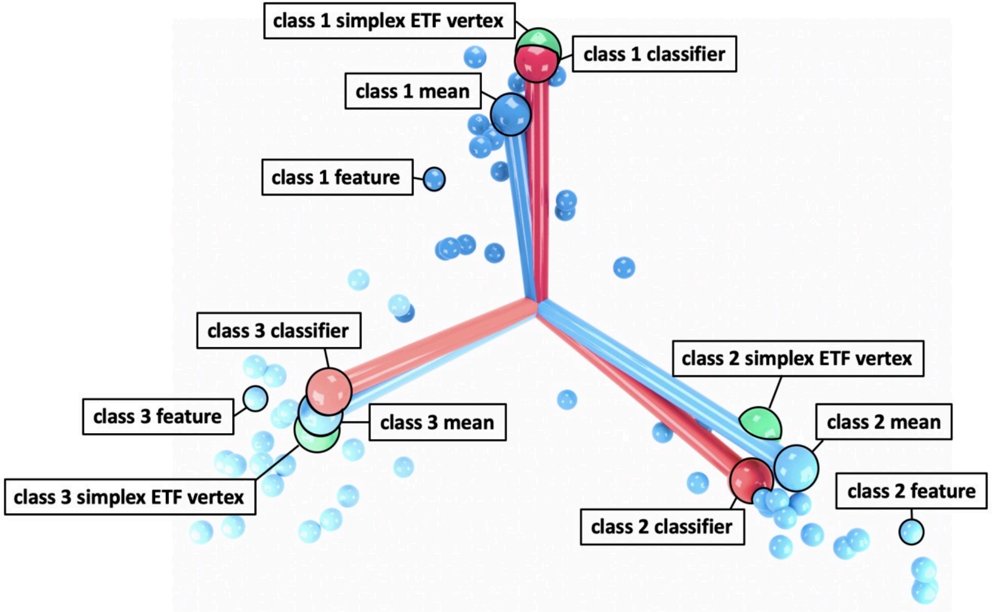


Image credited to Han et al. "Neural Collapse Under MSE Loss: Proximity to and Dynamics on the Central Path"

- Reveals common outcome of network training **across a variety of architectures** (ResNet, VGG) and **dataset** (CIFAR, ImageNet)
- **Precise mathematical structures** within the features and classifier

Neural Collapse: Symmetry and Structures

Balanced training dataset with $n = n_1 = n_2 = \cdots = n_K$, and

$$\boldsymbol{W} := \boldsymbol{W}_L, \quad \boldsymbol{H} := \begin{bmatrix} \boldsymbol{h}_{1,1} & \cdots & \boldsymbol{h}_{K,n} \end{bmatrix}.$$

Neural Collapse (NC) means that

1) ***Within-Class Variability Collapse on H:*** features of each class collapse to class-mean with **zero** variability;

$$\boldsymbol{h}_{k,i} \to \overline{\boldsymbol{h}}_k, \quad \forall k \in [K], \; i \in [n].$$

2) ***Convergence to Simplex ETF on H:*** the class means are **linearly separable**, and **maximally distant*****;***

$$\boldsymbol{M}^\top \boldsymbol{M} = \frac{K}{K-1}\left(\boldsymbol{I}_K - \frac{1}{K}\boldsymbol{1}_K\boldsymbol{1}_K^\top\right), \quad \boldsymbol{M} = \alpha \boldsymbol{U}\overline{\boldsymbol{H}}$$

Neural Collapse: Symmetry and Structures

Balanced training dataset with $n = n_1 = n_2 = \cdots = n_K$, and

$$\boldsymbol{W} := \boldsymbol{W}_L, \quad \boldsymbol{H} := \begin{bmatrix} \boldsymbol{h}_{1,1} & \cdots & \boldsymbol{h}_{K,n} \end{bmatrix}.$$

Neural Collapse (NC) means that

3) ***Convergence to Self-Duality (W,H):*** the last-layer classifiers are perfected matched with the class-means of features.

$$\boldsymbol{w}^k = \beta \overline{\boldsymbol{h}}_k, \quad \forall\, k \in [K].$$

4) ***Simple Decision Rule*** via Nearest Class-Center decision.

Simplification: Unconstrained Features

$$\psi_{\boldsymbol{\Theta}}(\boldsymbol{x}) \;=\; \overset{\text{Last-layer classifier}}{\boldsymbol{W}_L}\, \underbrace{\sigma\left(\boldsymbol{W}_{L-1} \cdots \sigma(\boldsymbol{W}_1 \boldsymbol{x} + \boldsymbol{b}_1) + \boldsymbol{b}_{L-1}\right)}_{\phi_{\boldsymbol{\theta}}(\boldsymbol{x}) =: \boldsymbol{h}} + \boldsymbol{b}_L$$

Last-layer classifier

Last-layer feature

Treat $\boldsymbol{H} = \begin{bmatrix} \boldsymbol{h}_{1,1} & \cdots & \boldsymbol{h}_{K,n} \end{bmatrix}$ as a free optimization variable

Simplification: Unconstrained Features

$$\psi_{\Theta}(\boldsymbol{x}) \;=\; \boldsymbol{W}_L \underbrace{\sigma\left(\boldsymbol{W}_{L-1}\cdots\sigma(\boldsymbol{W}_1\boldsymbol{x}+\boldsymbol{b}_1)+\boldsymbol{b}_{L-1}\right)}_{\phi_{\boldsymbol{\theta}}(\boldsymbol{x}) =: \boldsymbol{h}} + \boldsymbol{b}_L$$

Last-layer classifier ($\boldsymbol{W}_L$)

Last-layer feature ($\phi_{\boldsymbol{\theta}}(\boldsymbol{x}) =: \boldsymbol{h}$)

Treat $\boldsymbol{H} = \begin{bmatrix} \boldsymbol{h}_{1,1} & \cdots & \boldsymbol{h}_{K,n} \end{bmatrix}$ as a free optimization variable

$$\boxed{\min_{\boldsymbol{W},\boldsymbol{H},\boldsymbol{b}} \frac{1}{Kn}\sum_{k=1}^{K}\sum_{i=1}^{n}\mathcal{L}_{\mathrm{CE}}(\boldsymbol{W}\boldsymbol{h}_{k,i}+\boldsymbol{b},\boldsymbol{y}_k) + \frac{\lambda_{\boldsymbol{W}}}{2}\|\boldsymbol{W}\|_F^2 + \frac{\lambda_{\boldsymbol{H}}}{2}\|\boldsymbol{H}\|_F^2 + \frac{\lambda_{\boldsymbol{b}}}{2}\|\boldsymbol{b}\|_2^2}$$

Simplification: Unconstrained Features

$$\psi_{\boldsymbol{\Theta}}(\boldsymbol{x}) \;=\; \boldsymbol{W}_L \underbrace{\sigma\left(\boldsymbol{W}_{L-1}\cdots\sigma(\boldsymbol{W}_1\boldsymbol{x}+\boldsymbol{b}_1)+\boldsymbol{b}_{L-1}\right)}_{\phi_{\boldsymbol{\theta}}(\boldsymbol{x})=:\boldsymbol{h}} + \boldsymbol{b}_L$$

Last-layer classifier

Last-layer feature

Treat $\boldsymbol{H} = \begin{bmatrix} \boldsymbol{h}_{1,1} & \cdots & \boldsymbol{h}_{K,n} \end{bmatrix}$ as a free optimization variable

$$\boxed{\min_{\boldsymbol{W},\boldsymbol{H},\boldsymbol{b}} \frac{1}{Kn}\sum_{k=1}^{K}\sum_{i=1}^{n}\mathcal{L}_{\mathrm{CE}}(\boldsymbol{W}\boldsymbol{h}_{k,i}+\boldsymbol{b},\boldsymbol{y}_k) + \frac{\lambda_{\boldsymbol{W}}}{2}\|\boldsymbol{W}\|_F^2 + \frac{\lambda_{\boldsymbol{H}}}{2}\|\boldsymbol{H}\|_F^2 + \frac{\lambda_{\boldsymbol{b}}}{2}\|\boldsymbol{b}\|_2^2}$$

- **Validity:** Modern network are highly **overparameterized**, that can **approximate any point** in the feature space [Shaham'18];
- **State-of-the-Art:** also called **Layer-Peeled Model** [Fang'21], existing work [E'20, Lu'20, Mixon'20, Fang'21] **only** studied global optimality conditions.

Prior Work on Unconstrained Features

- [Lu et al'20] studies the following one-example-per class model

$$\min_{\boldsymbol{H}} \frac{1}{K} \sum_{k=1}^{K} \mathcal{L}_{\mathrm{CE}}(\boldsymbol{h}_k, \boldsymbol{y}_k), \ \text{s.t.} \quad \|\boldsymbol{h}_k\|_2 = 1$$

- [E et al'20] considers $\min_{\boldsymbol{W},\boldsymbol{H}} \frac{1}{Kn} \sum_{k=1}^{K} \sum_{i=1}^{n} \mathcal{L}_{\mathrm{CE}}(\boldsymbol{W}\boldsymbol{h}_{k,i}, \boldsymbol{y}_k), \ \text{s.t.} \quad \|\boldsymbol{W}\|_2 \le 1, \|\boldsymbol{h}_{k,i}\|_2 \le 1$
- [Fang et al'21] studies $\min_{\boldsymbol{W},\boldsymbol{H}} \frac{1}{Kn} \sum_{k=1}^{K} \sum_{i=1}^{n} \mathcal{L}_{\mathrm{CE}}(\boldsymbol{W}\boldsymbol{h}_{k,i}, \boldsymbol{y}_k), \ \text{s.t.} \quad \|\boldsymbol{W}\|_F^2 \le C_W, \|\boldsymbol{H}\|_F^2 \le C_H$
- These work show that any **global** solution has NC, but
  - What about local minima?
  - The constrain formulation are **not aligned with practice**
- [Mixon et al'21, Han et al'21] studies NC under the MSE loss

J. Lu and S. Steinerberger, Neural collapse with cross-entropy loss, 2020
W. E and S. Wojtowytsch, On the emergence of tetrahedral symmetry in the final and penultimate layers of neural network classifiers, 2020
D. Mixon, H. Parshall, J. Pi. Neural collapse with unconstrained features, 2020
C. Fang, H. He, Q. Long, W. Su, Layer-peeled model: Toward understanding well-trained deep neural networks, 2021
X. Han, V. Papyan, D. Donoho, Neural Collapse Under MSE Loss: Proximity to and Dynamics on the Central Path, 2021

Our Main Theoretical Results

Theorem (Informal) *Consider the nonconvex loss with unconstrained feature model with* $K < d$ and balanced data

$$\min_{\boldsymbol{W},\boldsymbol{H},\boldsymbol{b}} \frac{1}{Kn}\sum_{k=1}^{K}\sum_{i=1}^{n}\mathcal{L}_{\mathrm{CE}}(\boldsymbol{W}\boldsymbol{h}_{k,i}+\boldsymbol{b},\boldsymbol{y}_k)+\frac{\lambda_{\boldsymbol{W}}}{2}\|\boldsymbol{W}\|_F^2+\frac{\lambda_{\boldsymbol{H}}}{2}\|\boldsymbol{H}\|_F^2+\frac{\lambda_{\boldsymbol{b}}}{2}\|\boldsymbol{b}\|_2^2$$

- ***(Global Optimality)*** *Any global solution* $(\mathbf{W}_\star, \mathbf{H}_\star)$ *satisfies the NC properties (1-4).*
- ***(Benign Global Landscape)*** *The function has* ***no spurious*** *local minimizer and is a* ***strict saddle function****, with negative curvature for non-global critical point.*

UNIVERSITY OF MICHIGAN

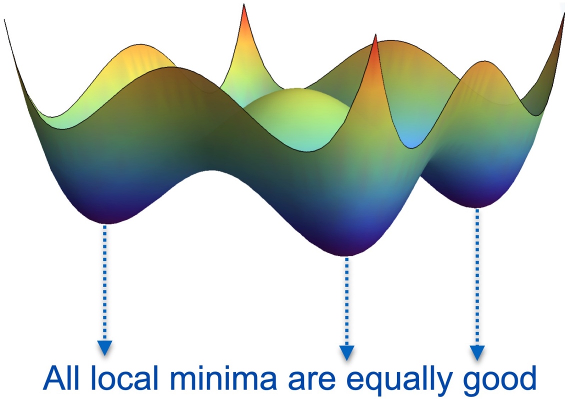
Our Main Theoretical Results

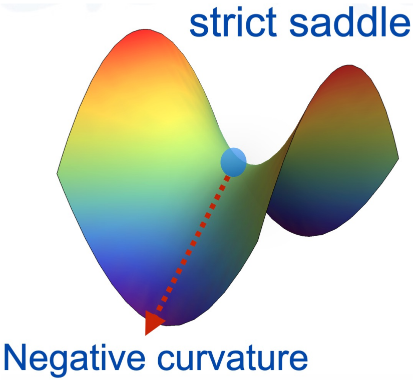
Theorem (Informal) *Consider the nonconvex loss with unconstrained feature model with* $K < d$ and balanced data

- ***(Global Optimality)*** *Any global solution* $(\mathbf{W}_\star, \mathbf{H}_\star)$ *satisfies the NC properties (1-4).*
- ***(Benign Global Landscape)*** *The function has* ***no spurious*** *local minimizer and is a* ***strict saddle function****, with negative curvature for nonglobal critical point.*

Message: deep networks always learn Neural Collapse features and classifiers, provably

Interpretations of Our Results





- **A Feature Learning Perspective.**
  - ➢ Top down: unconstrained feature model, representation learning, but no input information.
  - ➢ Bottom up: shallow network, strong assumptions, far from practice.
- **Connections to Empirical Phenomena.**

Interpretations of Our Results

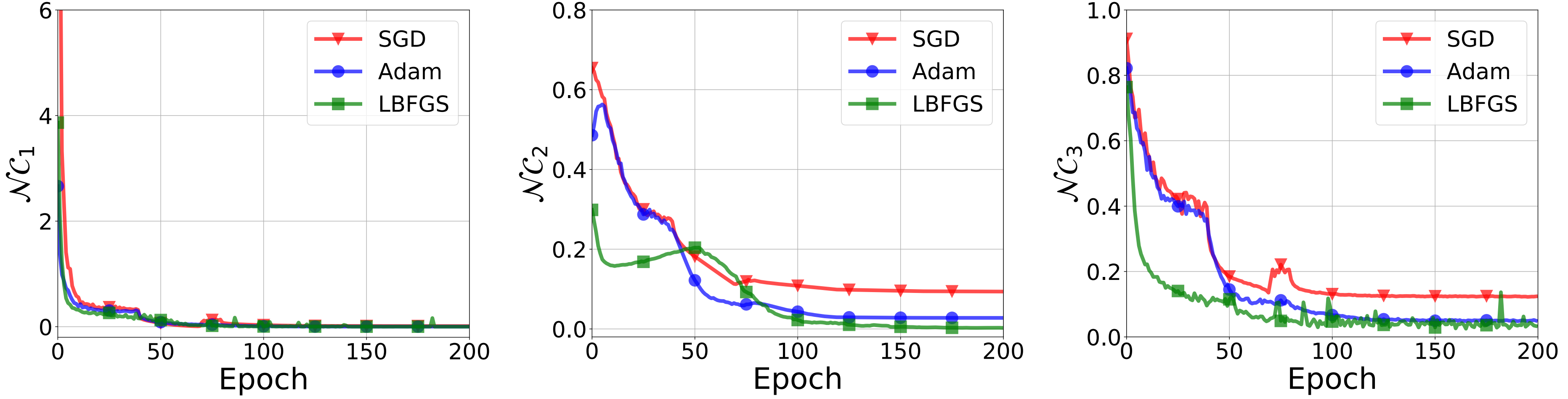
$$\boxed{\min_{\boldsymbol{W},\boldsymbol{H},\boldsymbol{b}} \frac{1}{Kn}\sum_{k=1}^{K}\sum_{i=1}^{n}\mathcal{L}_{\mathrm{CE}}(\boldsymbol{W}\boldsymbol{h}_{k,i}+\boldsymbol{b},\boldsymbol{y}_k)+\frac{\lambda_{\boldsymbol{W}}}{2}\|\boldsymbol{W}\|_F^2+\frac{\lambda_{\boldsymbol{H}}}{2}\|\boldsymbol{H}\|_F^2+\frac{\lambda_{\boldsymbol{b}}}{2}\|\boldsymbol{b}\|_2^2}$$

Closely relates to **low-rank matrix factorization** problems [Burer et al'03, Bhojanapalli et al'16, Ge et al'16, Zhu et al'18,Li et al'19, Chi et al'19]

- **Difference in tasks:** classification training vs recovery
- **Difference in global solutions**.
- **Difference in loss functions, statistical properties:** cross-entropy vs least-squares; randomness or statistical properties of the sensing matrices

Experiment: NC is Algorithm Independent

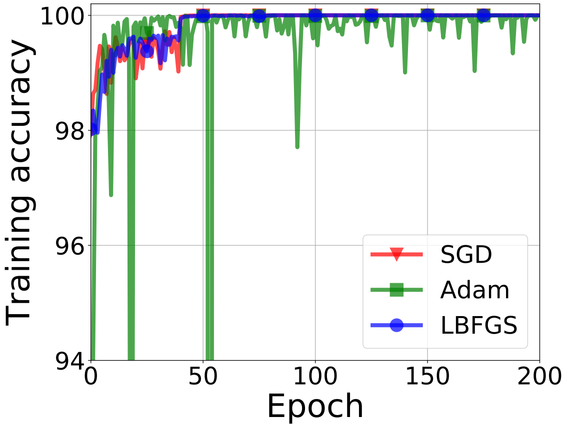
CIFAR-10 Dataset, ResNet18, with **different training algorithms**

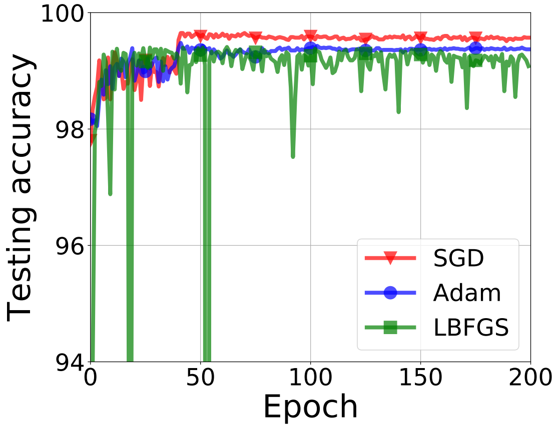


Measure of Within-Class Variability | Measure of Between-Class Separation | Measure of Self-Duality Collapse

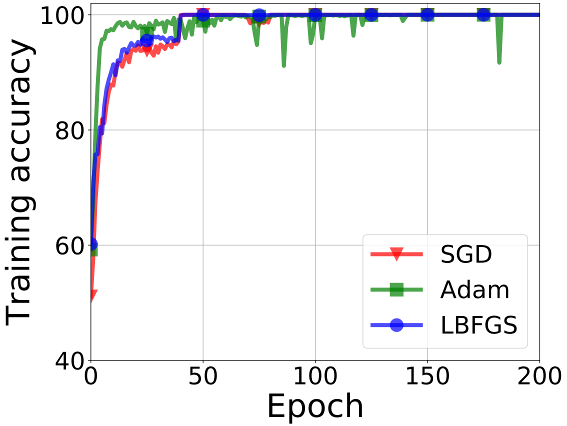
Generalization is Algorithm dependent

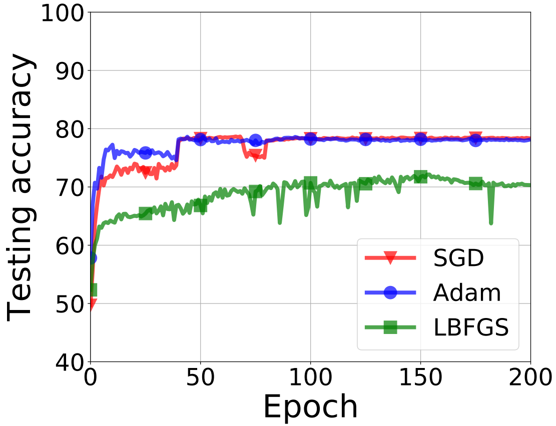
MINST





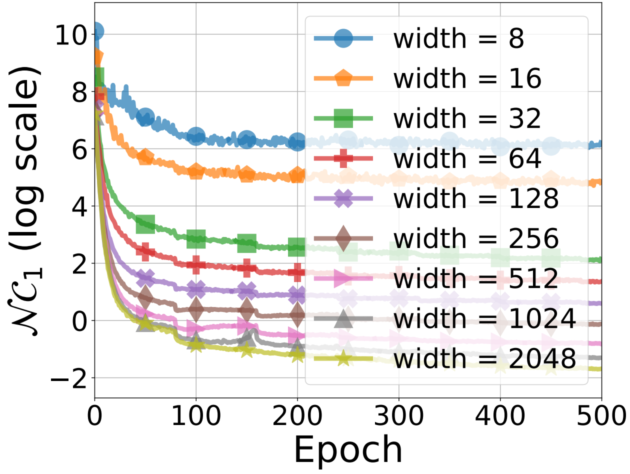
CIFAR-10



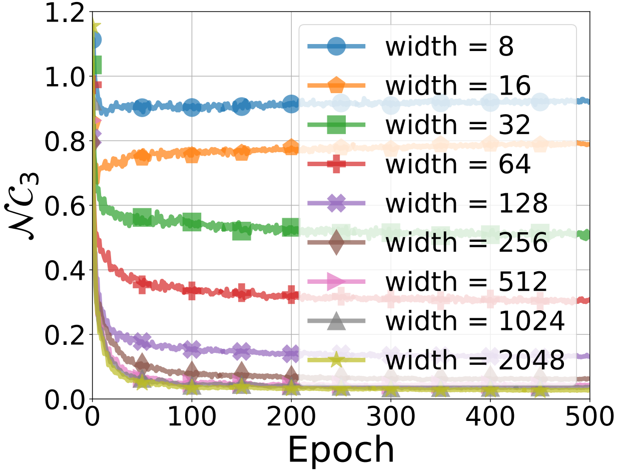


Experiment: NC Occurs for Random Labels

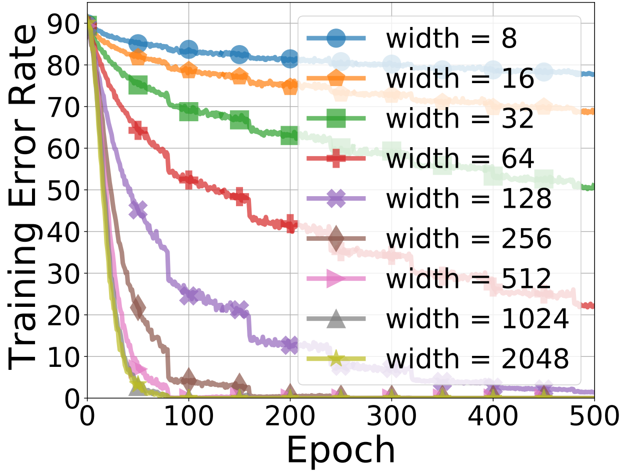
CIFAR-10 Dataset, MLP, **random labels with varying network width**



Measure of Within-Class Variability



Measure of Self-Duality Collapse

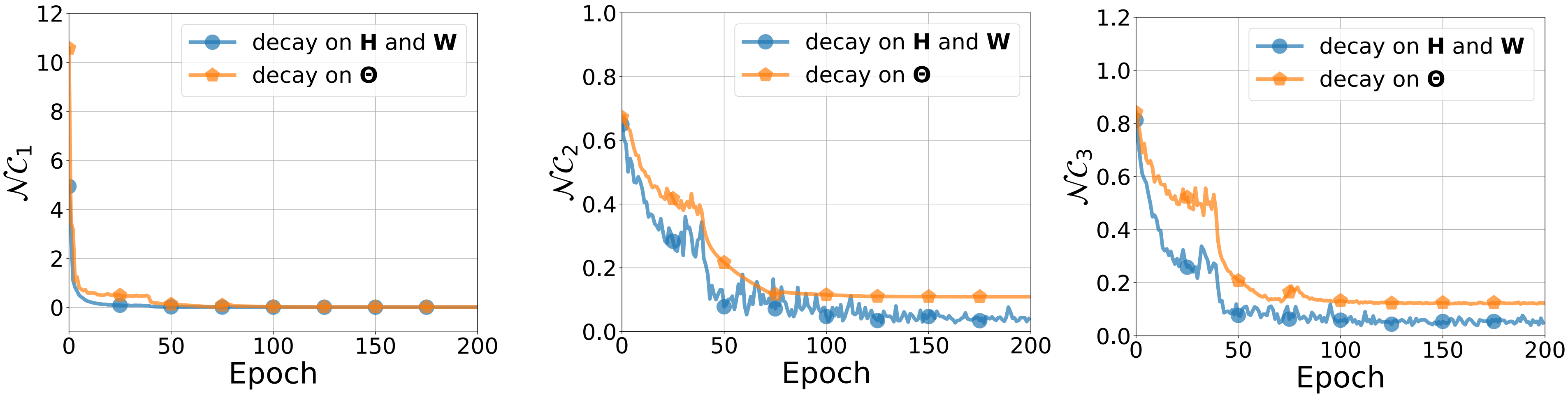


Training Error

Validity of Unconstrained Feature Model: Learned last-layer features and classifiers seems to be **independent of input!**

Experiment: NC with Different Weight Decays

CIFAR-10 Dataset, ResNet18, different weight decay



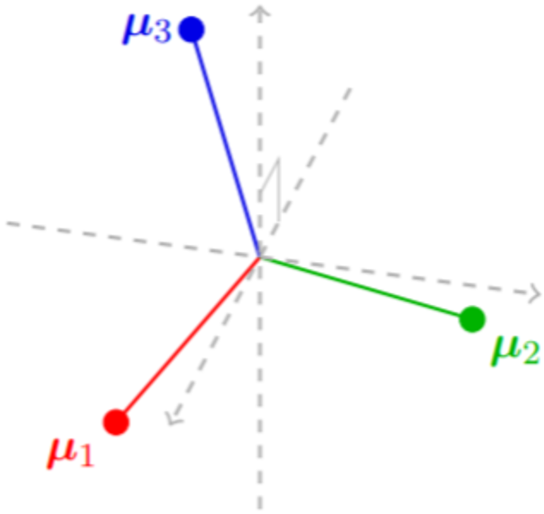
Measure of Within-Class Variability | Measure of Between-Class Separation | Measure of Self-Duality Collapse

- Test Accuracy: 99.57% vs. 99.60% (MINST); 77.92% vs. 78.42% (CIFAR-10)
- Weight decay on the parameters (implicitly) regularizes the features

Implications for Practical Network Training

Observation: For NC features, when $K \leq d$ the best classifier is given by the Simplex ETF

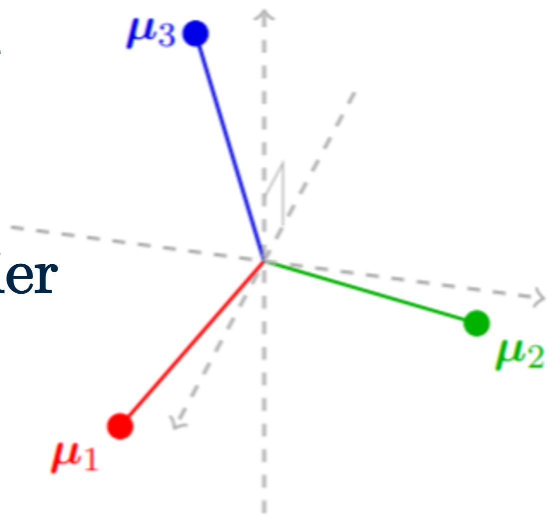
$$\boldsymbol{W}_\star = \begin{bmatrix} \boldsymbol{\mu}_1 & \cdots & \boldsymbol{\mu}_K \end{bmatrix}^\top.$$



Implications for Practical Network Training

Observation: For NC features, when $K \leq d$ the best classifier is given by the Simplex ETF

$$\boldsymbol{W}_\star = \begin{bmatrix} \boldsymbol{\mu}_1 & \cdots & \boldsymbol{\mu}_K \end{bmatrix}^\top.$$

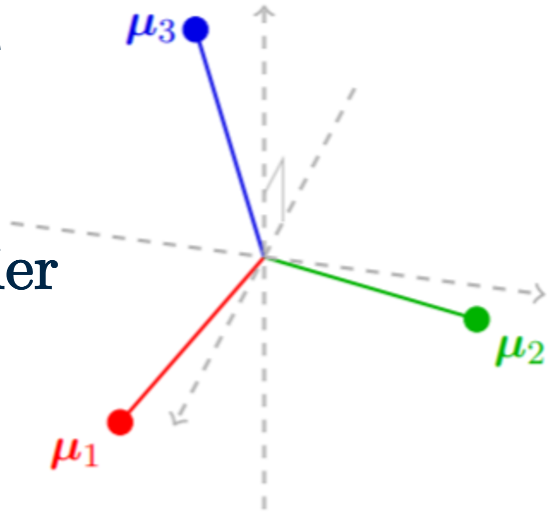


- **Implication 1: No need to learn the classifier**
  - Just fix them as a Simplex ETF
  - Save **8%, 12%,** and **53%** parameters for ResNet50, DenseNet169, and ShuffleNet!

Implications for Practical Network Training

Observation: For NC features, when $K \leq d$ the best classifier is given by the Simplex ETF

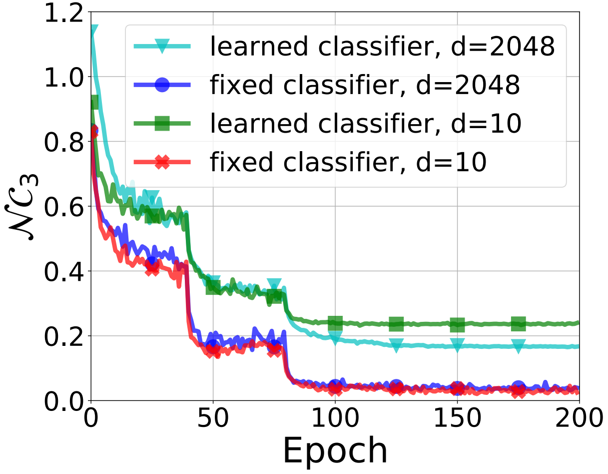
$$\boldsymbol{W}_\star = \begin{bmatrix} \boldsymbol{\mu}_1 & \cdots & \boldsymbol{\mu}_K \end{bmatrix}^\top .$$



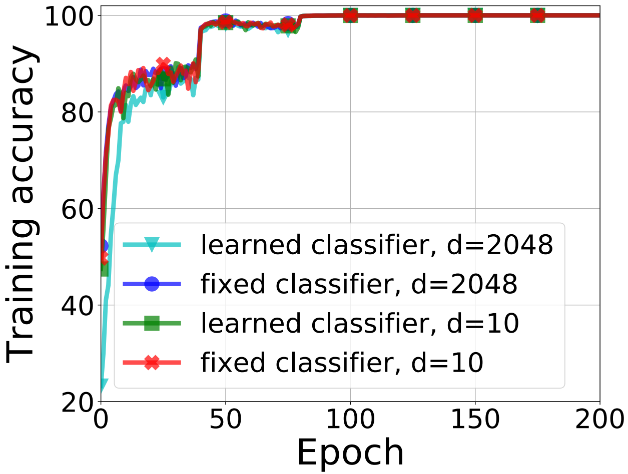
- **Implication 1: No need to learn the classifier**
  - ❑ Just fix them as a Simplex ETF
  - ❑ Save **8%, 12%,** and **53%** parameters for ResNet50, DenseNet169, and ShuffleNet!
- **Implication 2: No need of large feature dimension d**
  - ❑ Just use feature dim d = #class K (e.g., d=10 for CIFAR10)
  - ❑ Further saves **21%** and **4.5%** parameters for ResNet18 and ResNet50!

Experiment: Fixed Classifier with $d = K$

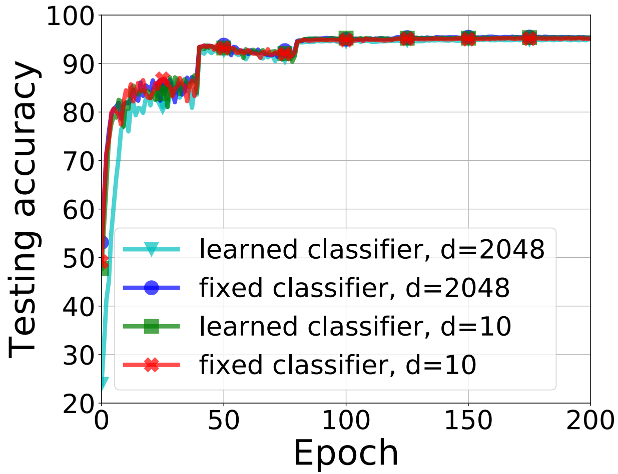
ResNet50, CIFAR10, Comparison of **Learned vs. Fixed Classifiers of W**



Measure of Between-Class Separation



Training Accuracy



Testing Accuracy

Training with fixed last-layer classifiers achieves **on-par performance** with learned classifiers.

Summary and Discussion

Z. Zhu, T. Ding, J. Zhou, X. Li, C. You, J. Sulam, and **Q. Qu**, A Geometric Analysis of Neural Collapse with Unconstrained Features, *arXiv Preprint arXiv:2105.02375*, May 2021.

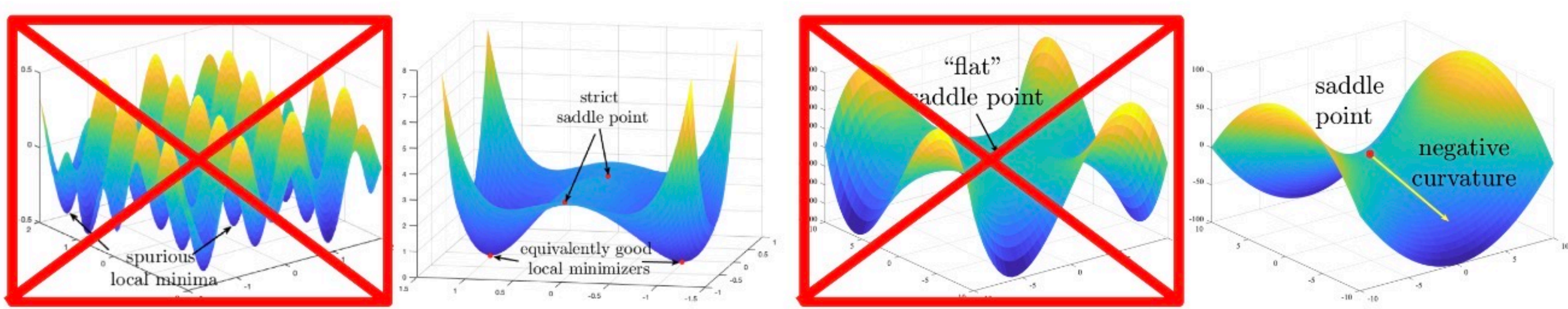
- Through landscape analysis under unconstrained feature model, we provide a **complete characterization of learned representation** of deep networks.

- The understandings of learned representations could shed lights on generalization, robustness, and transferability.

Outline of this Talk

- Learning Shallow Representations: (Convolutional) Dictionary Learning
- Learning Deep Representations: Deep Neural Collapse
- **Conclusion & Discussion**

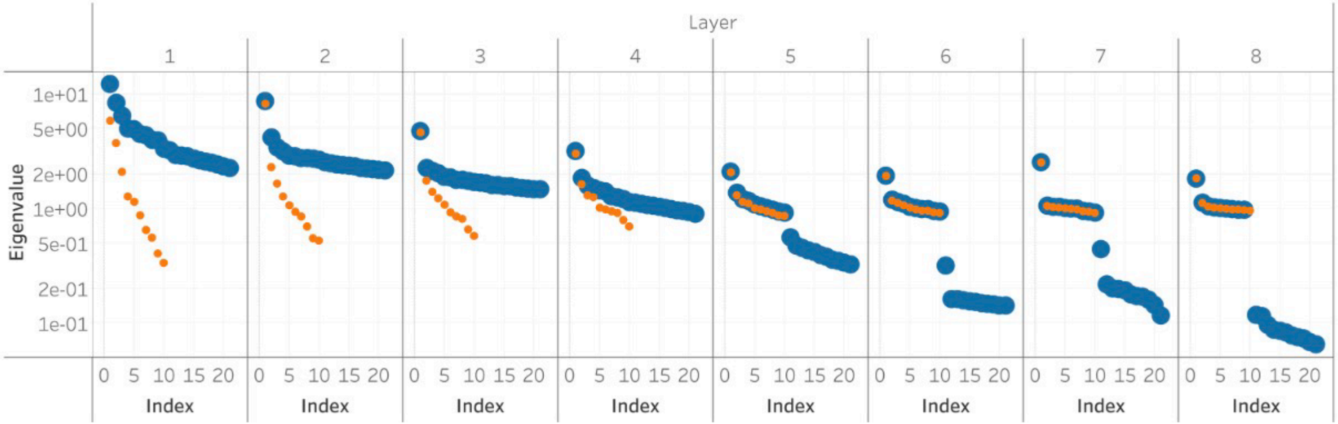
Summary and Discussion



1. **Q. Qu**, Y. Zhai, X. Li, Y. Zhang, Z. Zhu, Analysis of optimization landscapes for overcomplete learning, *ICLR'20*, (oral, top 1.9%)
2. Y. Lau (*), **Q. Qu**(*), H. Kuo, P. Zhou, Y. Zhang, J. Wright, Short-and-sparse Deconvolution – A Geometric Approach, *ICLR'20*
3. Z. Zhu, T. Ding, J. Zhou, X. Li, C. You, J. Sulam, and **Q. Qu**, A Geometric Analysis of Neural Collapse with Unconstrained Features, *arXiv Preprint arXiv:2105.02375*, May 2021.

Future Directions: Beyond Last-layer Features

- **Study Deeper Networks**
  - Fix the last layer classifier $\boldsymbol{W}$ as the Simplex ETF, and conduct NTK analysis for the learning dynamics of features $\boldsymbol{H}$?
  - Recursively study the features of each layer from output?

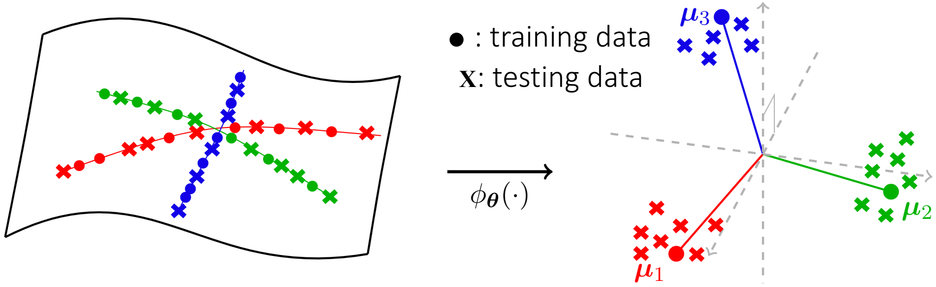


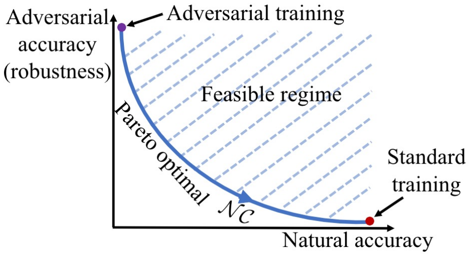
[Paypan'19]

Vardan Papyan. Traces of class/cross-class structure pervade deep learning spectra, JMLR'19.

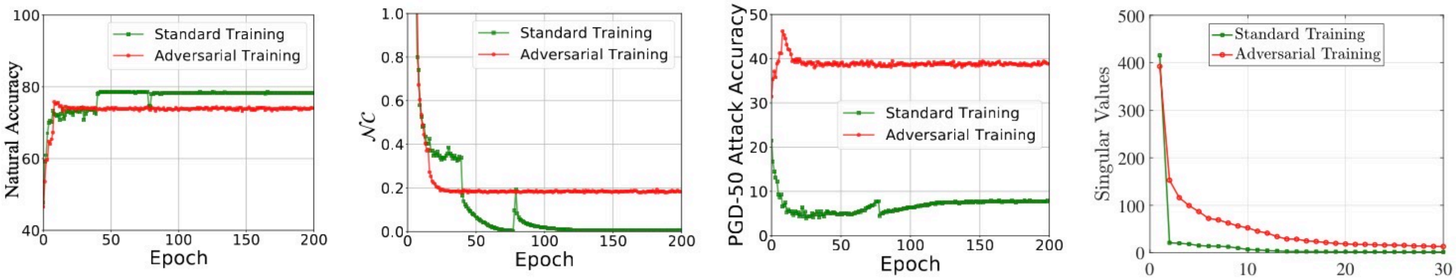
Future Directions: Is NC a Blessing or Curse?

- **Study generalization through the representation?**



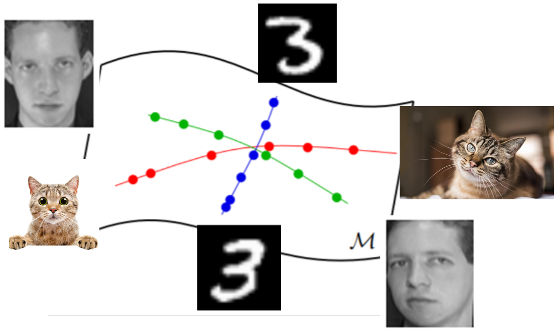


- **Study tradeoff between accuracy and robustness via NC?**

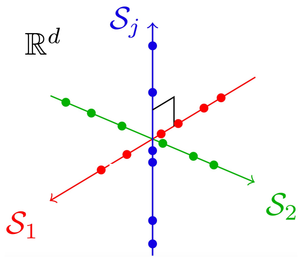


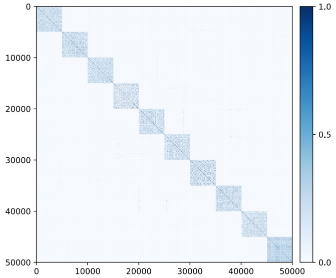
H. Zhang, Y. Yu, J. Jiao, E. Xing, L. Ghaoui, M. Jordan, Theoretically principled trade-off between robustness and accuracy, ICML2019.

Adaptive to the Intrinsic Data Structures

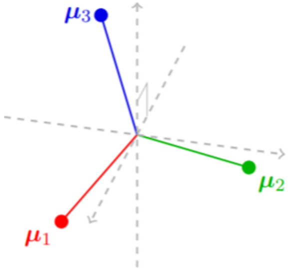


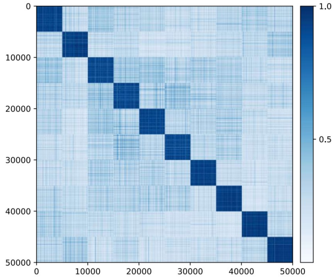
MCR²





Cross-entropy

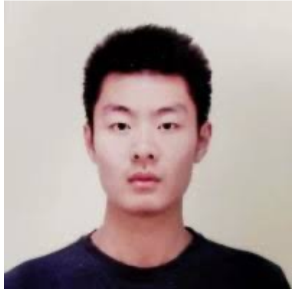




- Can we learn diverse features that are adaptive to the intrinsic data structures?

Acknowledgement

Tianyu Ding
(Johns Hopkins)



Xiao Li
(U. Michigan)

Xiao Li
(CUHK-Shen Zhen)

Jeremias Sulam
(Johns Hopkins)

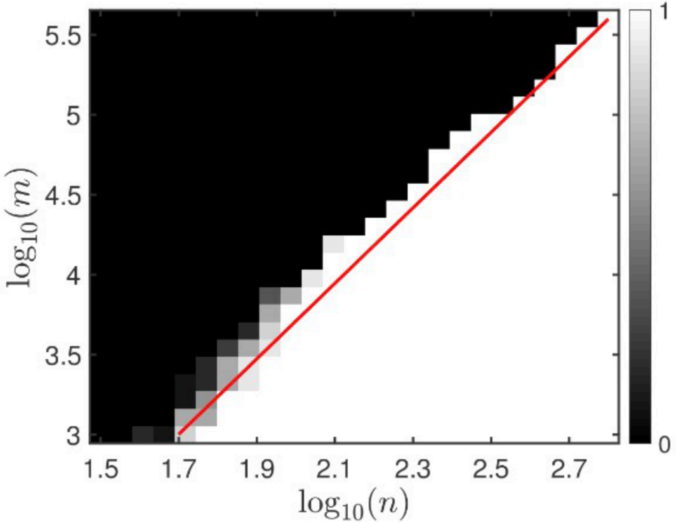
Chong You
(Google Research)

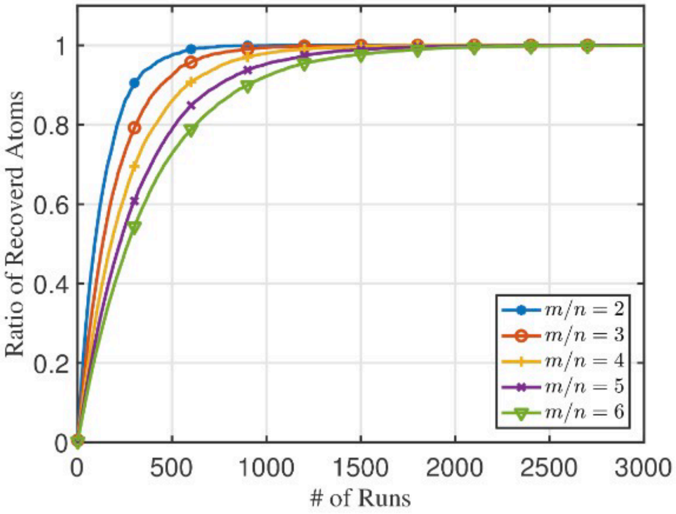
Yuexiang Zhai
(UC Berkeley)

Zhihui Zhu
(University of Denver)

Thank You!

Relationship to Dictionary Learning



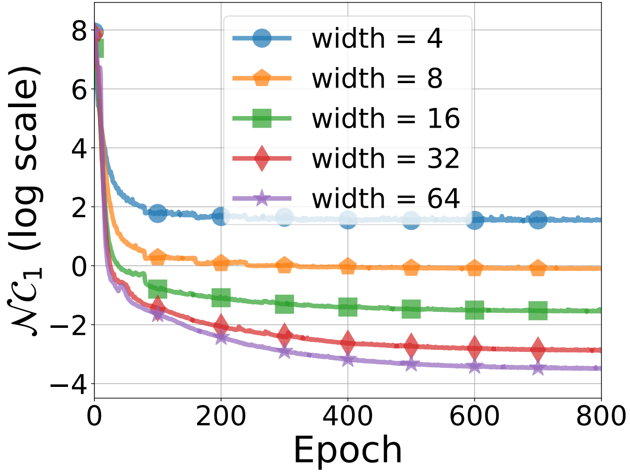


practice $m < n^2$
vs. theory $m < Cn$

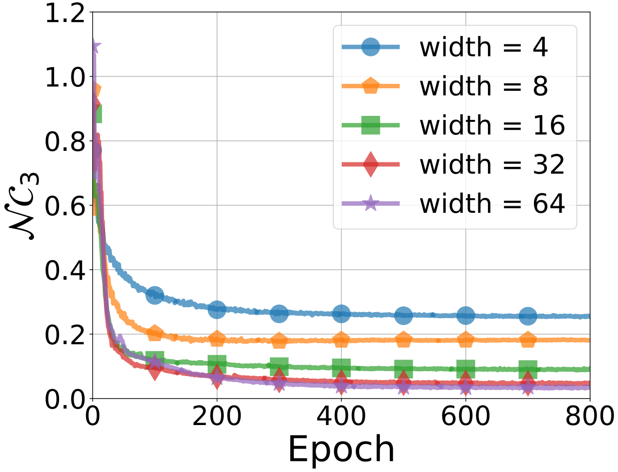
recover full $\boldsymbol{A}_0$ via repeated independent trials

Experiment: NC Occurs for Random Labels

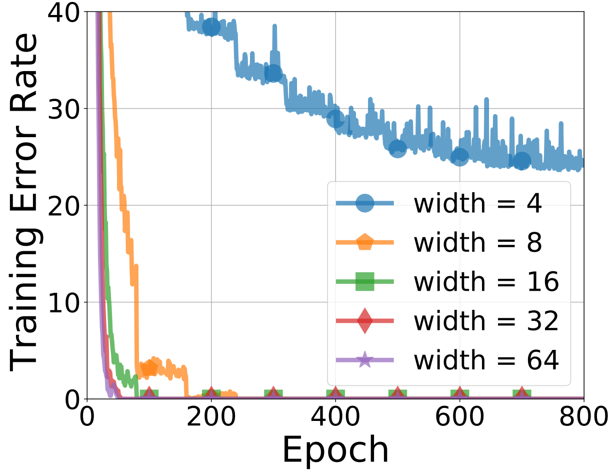
CIFAR-10 Dataset, ResNet18, random labels with varying network width



Measure of Within-Class Variability



Measure of Self-Duality Collapse



Training Error

Validity of Unconstrained Feature Model: Learned last-layer features and classifiers seems to be **independent of input!**

UNIVERSITY OF MICHIGAN

Comparisons to MCR^2

- [Yu et al, NeurIPS'20] learns not only discriminative but also diverse representations via maximizing the difference between the coding rate of all features and the average rate of features in the classes:

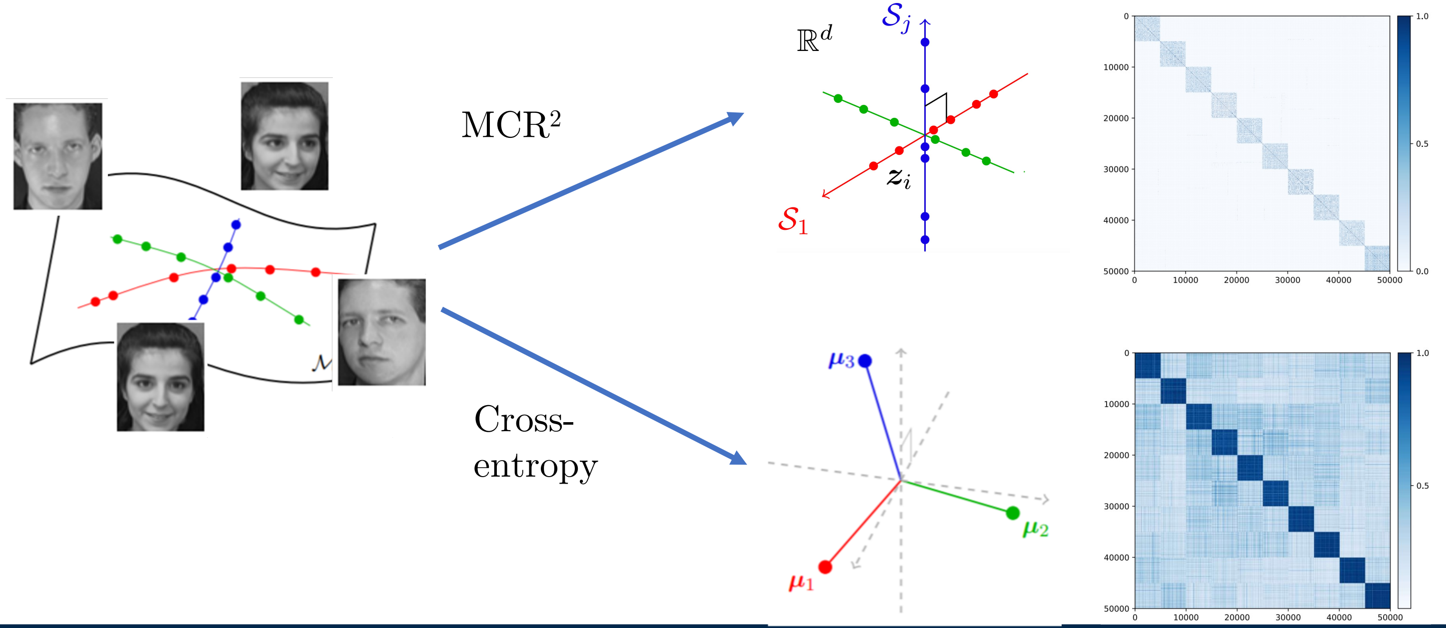
$$\Delta R(\boldsymbol{H},\epsilon) = \underbrace{\frac{1}{2}\log\det(\mathbf{I}+\frac{d}{n\epsilon^2})}_{R} - \underbrace{\sum_{k=1}^{K}\frac{n_k}{2n}\log\det(\mathbf{I}+\frac{d}{n_k\epsilon^2}\boldsymbol{H}_k\boldsymbol{H}_k^\top)}_{R^c}$$

  - R: expand all features $\boldsymbol{H}$ as large as possible.
  - R^c: compress all each class $\boldsymbol{H}_k$ as small as possible.
  - For balanced data, learned features $\boldsymbol{H}_k$ span an entire d/K subspace, and the subspaces are orthogonal to each other.

Y. Yu, K. Chan, C. You, C. Song, Y. Ma, Learning diverse and discriminative representations via the principle of maximal coding rate reduction, NeurIPS 20.
K. Chan, Y. Yu, C. You, H. Qi, J.Wright, and Y. Ma, ReduNet: A White-box Deep Network from the Principle of Maximizing Rate Reduction, 2021.

Comparisons to MCR²

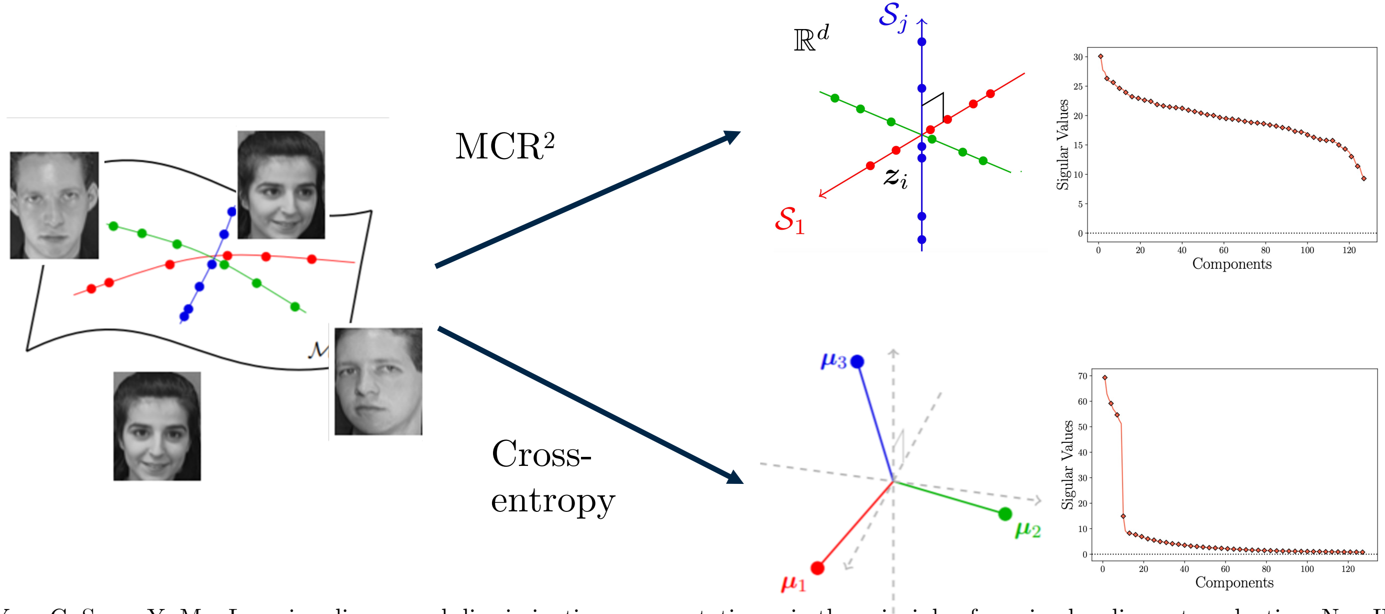
- [Yu et al, NeurIPS'20] learns not only discriminative but also diverse representations via maximizing the difference between the coding rate of all features and the average rate of features in the classes:



Y. Yu, K. Chan, C. You, C. Song, Y. Ma, Learning diverse and discriminative representations via the principle of maximal coding rate reduction, NeurIPS 20
K. Chan, Y. Yu, C. You, H. Qi, J.Wright, and Y. Ma, ReduNet: A White-box Deep Network from the Principle of Maximizing Rate Reduction, 2021.

Comparisons to MCR²

- [Yu et al, NeurIPS'20] learns not only discriminative but also diverse representations via maximizing the difference between the coding rate of all features and the average rate of features in the classes:



Y. Yu, K. Chan, C. You, C. Song, Y. Ma, Learning diverse and discriminative representations via the principle of maximal coding rate reduction, NeurIPS 20.
K. Chan, Y. Yu, C. You, H. Qi, J.Wright, and Y. Ma, ReduNet: A White-box Deep Network from the Principle of Maximizing Rate Reduction, 2021.