# Learning Deep Low-Dimensional Models from High-Dimensional Data: From Theory to Practice

## Lecture 2-2: Emergency of Low-dimensional Representations in Deep Models

Zhihui Zhu

The Ohio State University

THE OHIO STATE UNIVERSITY

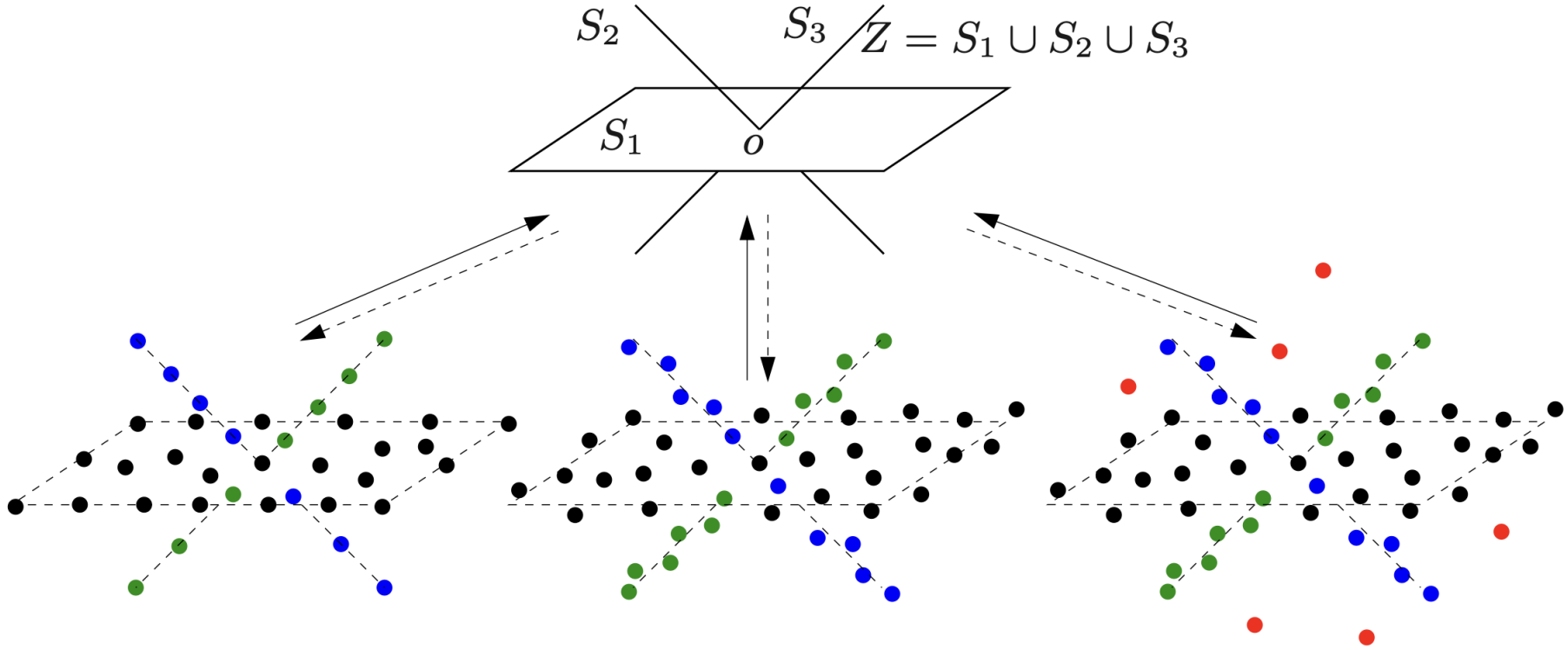
Oct 19, 2025, ICCV

# This Tutorial: The Outline

- Session I: Introduction of Basic Low-dimensional Models
- Session II: Understanding Low-Dimensional Structures in Representation Learning
  - Lec 2.1: Bridging Symbolic Abstraction and Low-Dimensionality in Machine Reasoning: Algebraic and Geometric Perspectives
  - **Lec 2.2: Emergency of Low-dimensional Representations in Deep Models**
- Session III: Understanding Low-Dimensional Structures in Diffusion Generative Models
  - Lec 3.1: Low-Dimensional Models for Understanding Generalization in Diffusion Models
  - Lec 3.2: Explore Low-Dimensional Structures for Constrained and Controllable Diffusion Models in Scientific Applications
- Session IV: Designing Deep Networks for Pursuing Low-Dimensional Structures
  - Lec 4.1: ReduNet: A White-box Deep Network from the Principle of Maximizing Rate Reduction
  - Lec 4.2: White-Box Transformers via Sparse Rate Reduction
- Session V: Panel Discussion: Sara Fridovich-Keil, Berivan Isik, Vladimir Pavlovic

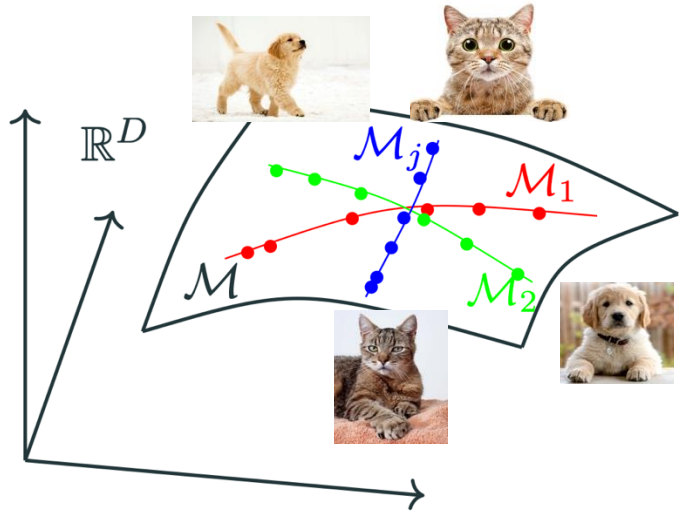
# Classical Low-dimension Model: GPCA

- Generalized PCA for mixture of subspaces [Vidal, Ma, Sastry 2005]

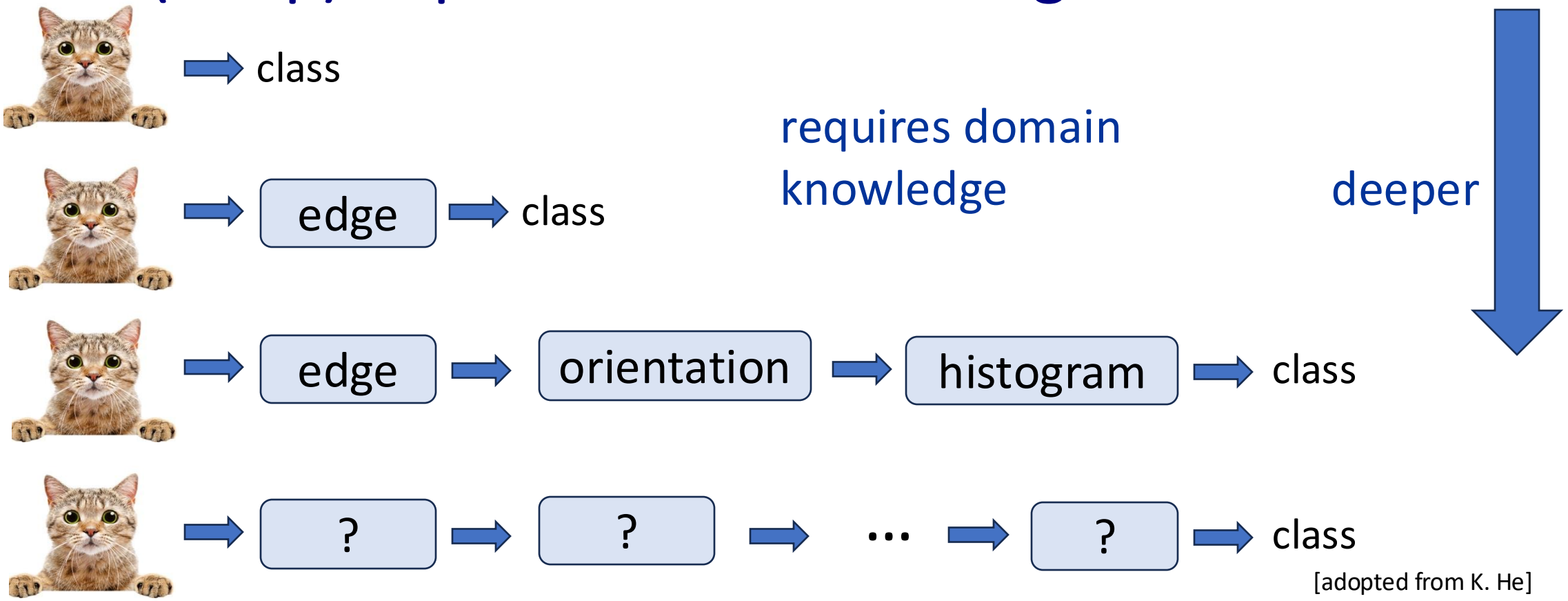


# Modern (Deep) Representation Learning

Understand and interact with the physical world $\Longrightarrow$ *nonlinear data*
Coping with nonlinearity demands (*deeper*) *representation*

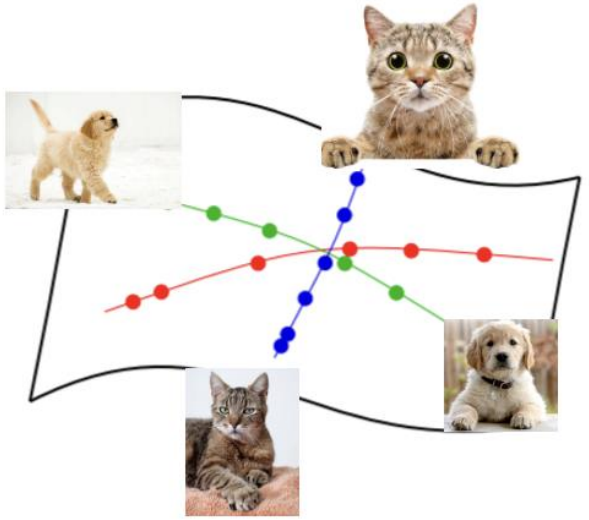


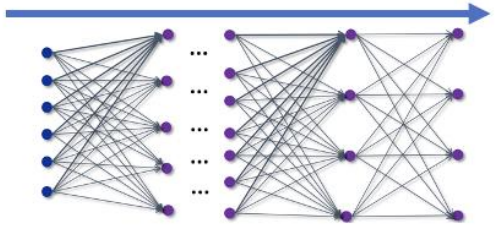
# Modern (Deep) Representation Learning

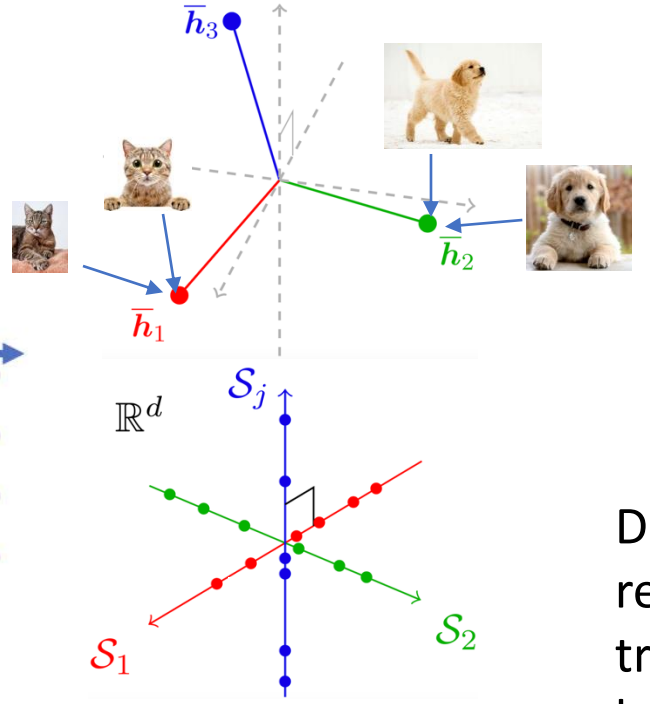



- Deep learning builds multiple level of abstractions
  - Learn representation from data by back-propagation
  - Reduce domain knowledge and feature engineering
  - Progressively "*linearize*" the nonlinear structure

**The objective of learning**:
Transform nonlinear and complex data to
a linear, compact, and structured representation.





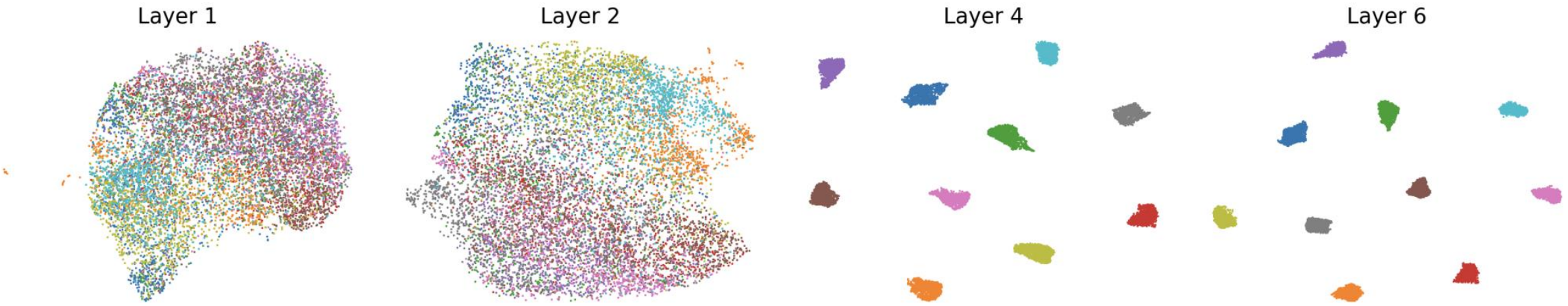

Neural collapse representation (by standard CE training)

Diverse & discriminative representation (by MCR$^2$ training, contrastive learning)

- Empirically observe across many architectures and dataset
- Theoretically justify for a simple model
- Lead to principled ways for designing architectures to pursue Low-D structures (see Session IV, afternoon)

**The effect of depth**:

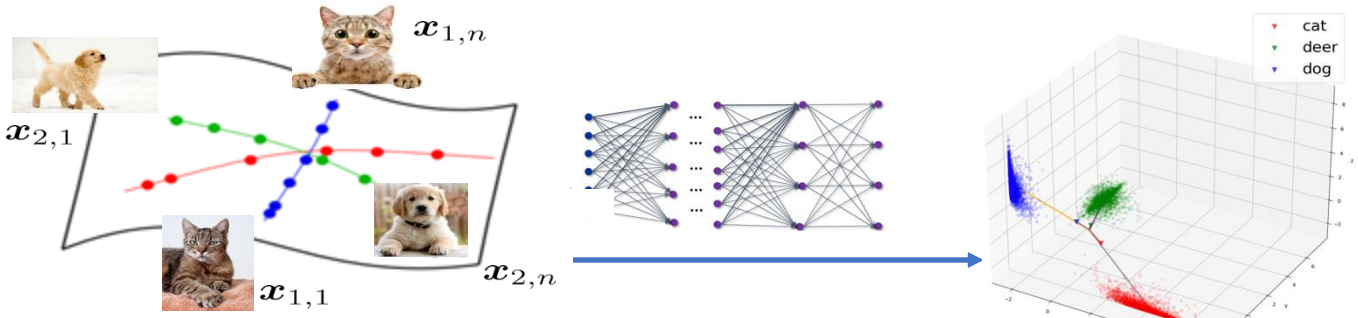
Progressively transform a nonlinear and complex distribution to a linear, compact and structured one




- Empirically observe across many architectures and dataset
- Theoretically justify for simplified models or geodesic curve connecting the two probability distributions
- Lead to principled ways for designing architectures to pursue Low-D structures

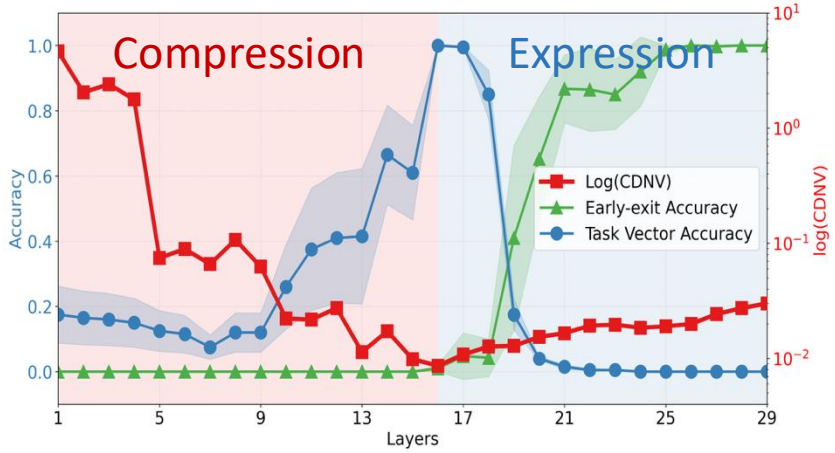
# Outline

- Low-D representation in Deep Classifiers:
  - Neural Collapse (NC) Phenomena
  - Understanding NC from optimization
  - Progressive NC and transferability




Data in the input space

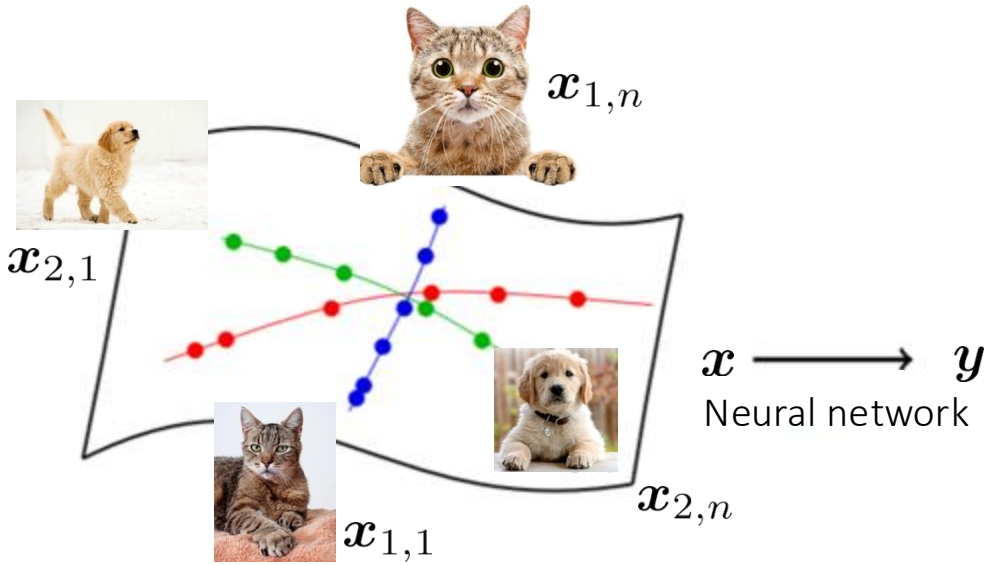
- Low-D representation in Generative Models (MLLM)
  - Compression-to-Expression Phenomena



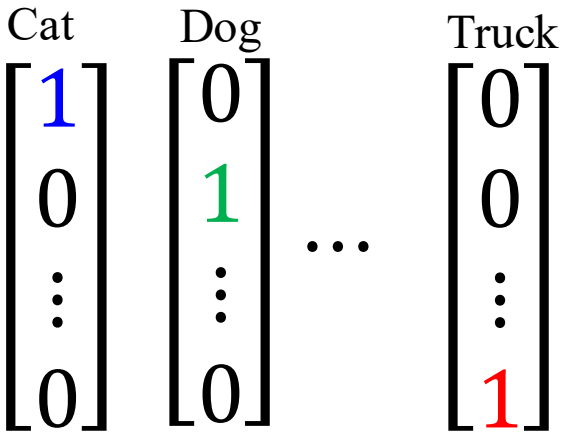

- Conclusion

# Multi-Class Image Classification Problems

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



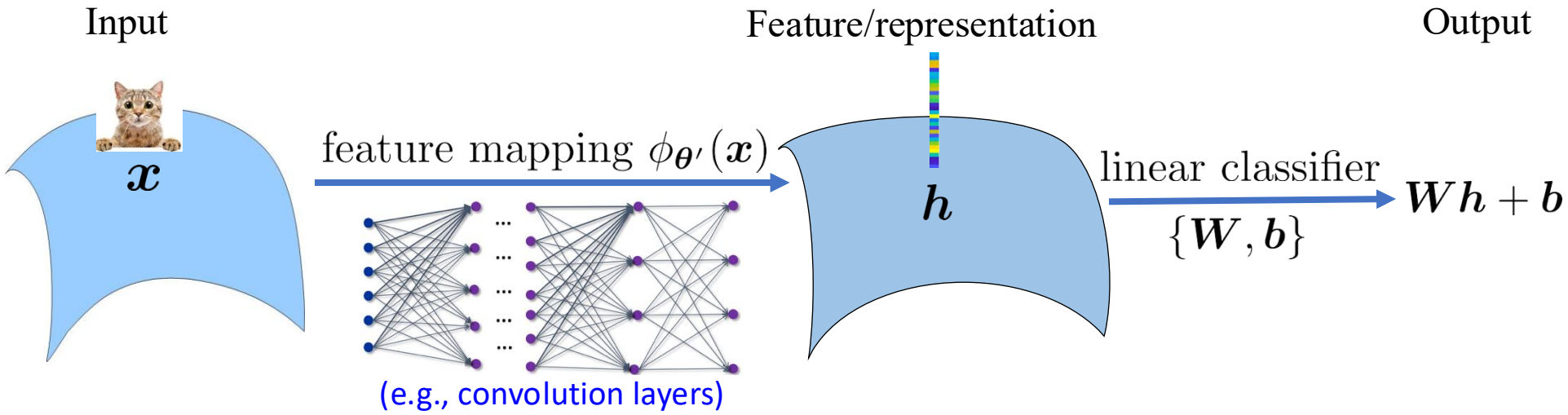

Data in the input space




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

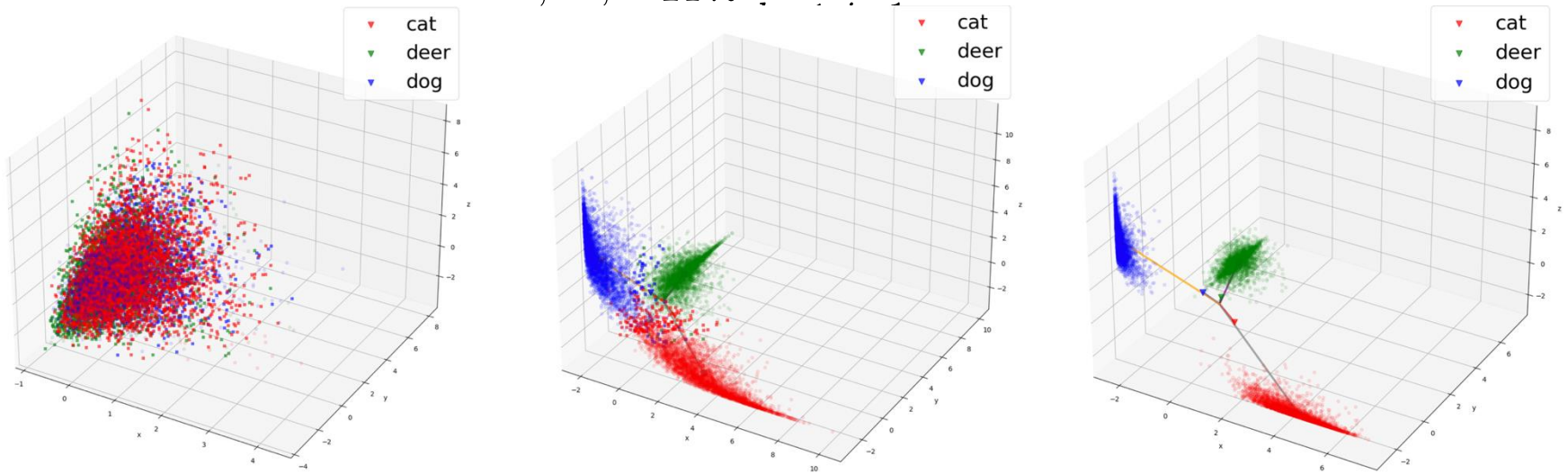
- Assume balanced dataset where each class has $n$ training samples

# Deep Neural Network Classifiers




- Training a deep neural network $\min_{\boldsymbol{\theta}', \boldsymbol{W}, \boldsymbol{b}} \frac{1}{Kn} \sum^{K} \sum^{n} \mathcal{L}_{\mathrm{CE}} \left( \boldsymbol{W} \phi_{\boldsymbol{\theta}'}(\boldsymbol{x}_{k,i}) + \boldsymbol{b}, \boldsymbol{y}_k \right)$

Features (dim = 3) of CIFAR10 learned by ResNet18




Initialization middle stage final stage

# Representation: Neural Collapse

- Characterize the geometry of representations for data separation

- [Papyan et al. 20] reveals common outcome of learned features (output of ***last-hidden*** layer) and classifiers across a variety of architectures and dataset

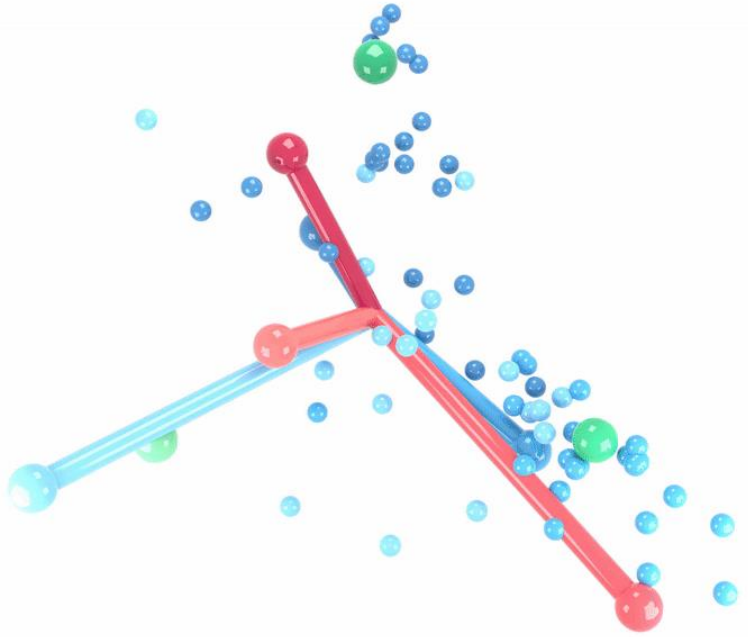
- Precise mathematical structure within the features and classifier

**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

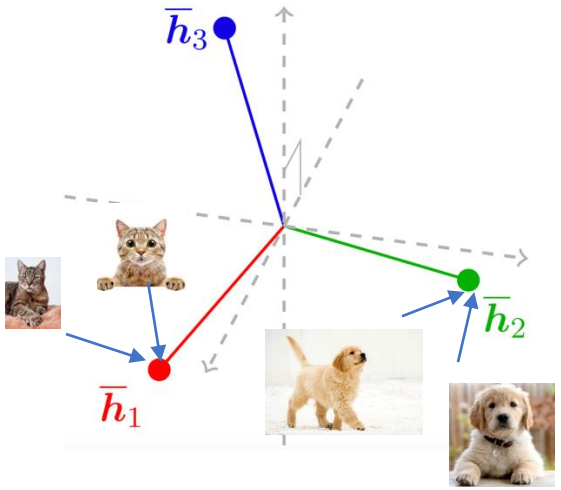
Contributed by David L. Donoho, August 18, 2020 (sent for review July 22, 2020; reviewed by Helmut Boelscker and Stéphane Mallat)



# Neural Collapse: Symmetry and Structures

- **Within-Class Variability Collapse (NC1)**: features of each class collapse to class-mean with zero variability (low-dimensional features):

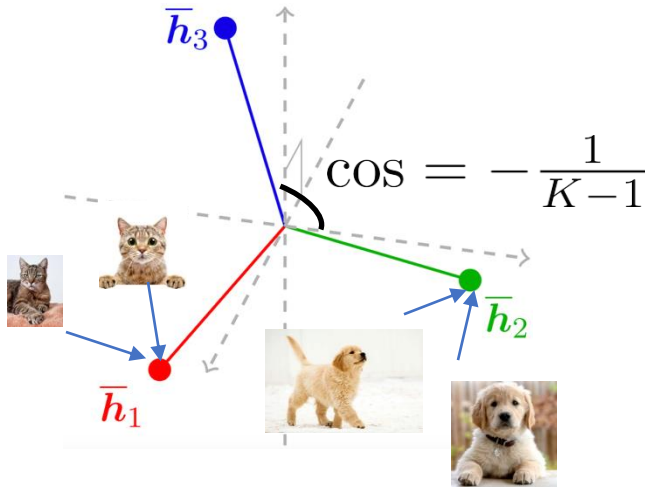
$$k\text{-th class}, i\text{-th sample} : \boldsymbol{h}_{k,i} \rightarrow \overline{\boldsymbol{h}}_k$$

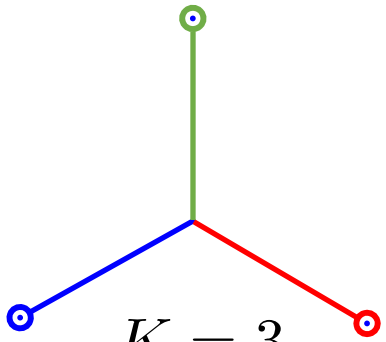


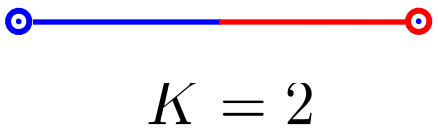
# Neural Collapse: Symmetry and Structures

- **Convergence to Simplex ETF (NC2)**: the class means are linearly separable, have same length, and maximal angle between each other:

$$\frac{\langle \overline{\boldsymbol{h}}_k, \overline{\boldsymbol{h}}_{k'} \rangle}{\|\overline{\boldsymbol{h}}_k\| \|\overline{\boldsymbol{h}}_{k'}\|} \to \begin{cases} 1, & k = k' \\ -\frac{1}{K-1}, & k \neq k' \end{cases}$$



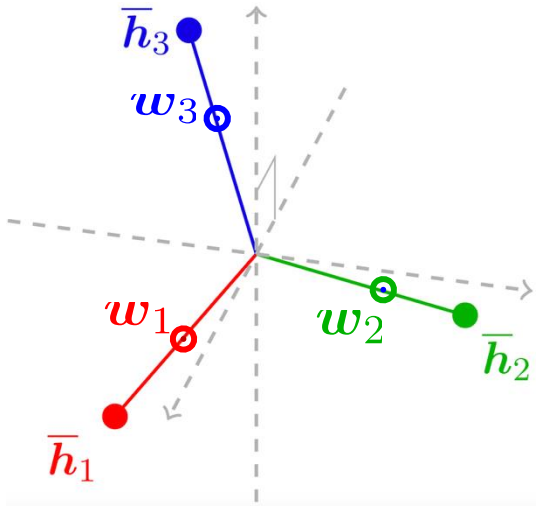



# Neural Collapse: Symmetry and Structures

- **Convergence to Self-Duality (NC3)**: the last-layer classifiers are perfectly matched with the class-means of features:

$$\frac{\boldsymbol{w}_k}{\|\boldsymbol{w}_k\|} \to \frac{\boldsymbol{h}_k}{\|\overline{\boldsymbol{h}}_k\|};$$

where $\boldsymbol{w}_k$ represents the $k$-th row of $\boldsymbol{W}$

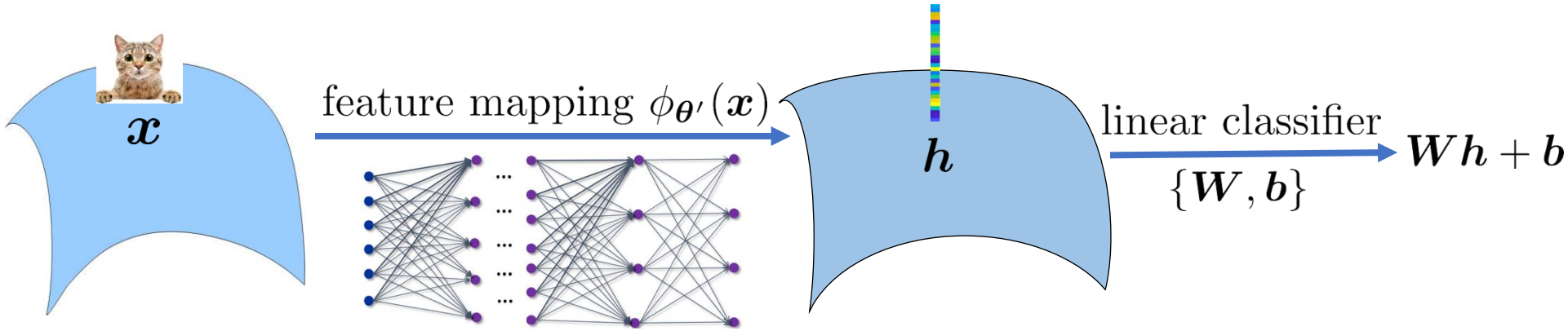


# Why Neural Collapse

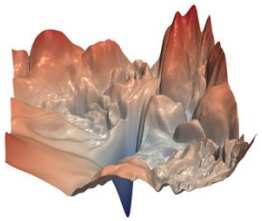
- NC is preferred among every successful exercise in feature engineering
  - Information Theory: Simplex ETF is the optimal Shannon code
  - Classification: Simple ETF features ⇒ Simplex ETF max-margin classifier

**Question**: Given the prevalence of Neural Collapse across datasets and network architectures, why would such a phenomenon happen in training overparameterized networks?

# Dealing with a Highly Nonconvex Problem



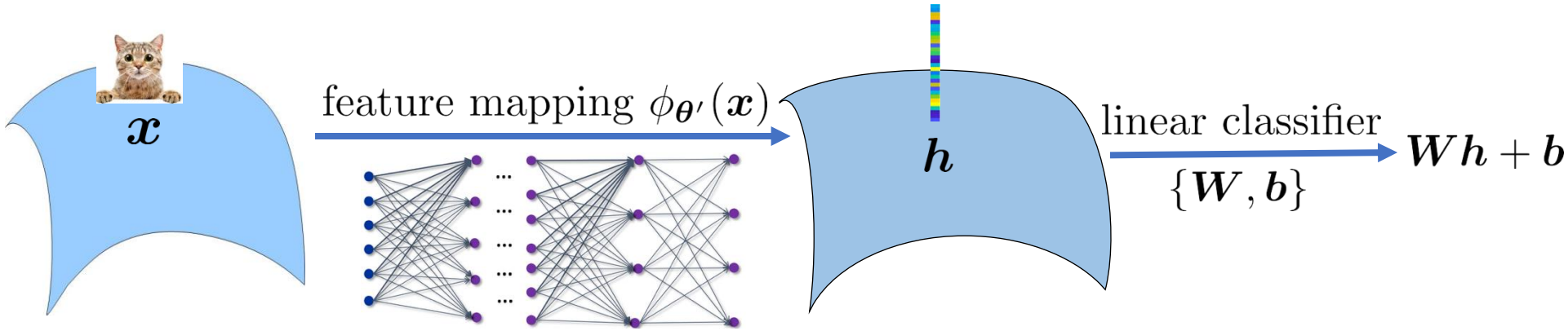

$$\min_{\boldsymbol{\theta}', \boldsymbol{W}, \boldsymbol{b}} \frac{1}{Kn} \sum_{k=1}^{K} \sum_{i=1}^{n} \mathcal{L}_{\mathrm{CE}} \left( \boldsymbol{W} \phi_{\boldsymbol{\theta}'}(\boldsymbol{x}_{k,i}) + \boldsymbol{b}, \boldsymbol{y}_k \right) + \lambda \left\| (\boldsymbol{\theta}', \boldsymbol{W}, \boldsymbol{b}) \right\|_F^2$$



The problem is highly nonconvex [Li et al'18]

- Nonlinear interactions across layers
- Nonlinear activation functions

# Simplification: Unconstrained Features




$$\min_{\boldsymbol{\theta}', \boldsymbol{W}, \boldsymbol{b}} \frac{1}{Kn} \sum_{k=1}^{K} \sum_{i=1}^{n} \mathcal{L}_{\mathrm{CE}} \left( \boldsymbol{W} \phi_{\boldsymbol{\theta}'}(\boldsymbol{x}_{k,i}) + \boldsymbol{b}, \boldsymbol{y}_k \right) + \lambda \left\| (\boldsymbol{\theta}', \boldsymbol{W}, \boldsymbol{b}) \right\|_F^2$$

- Assumption

Treat $\boldsymbol{h}_{k,i} = \phi_{\boldsymbol{\theta}'}(\boldsymbol{x}_{k,i})$ as a **free** optimization variable

Unconstrained features model
[Mixon et al'20]

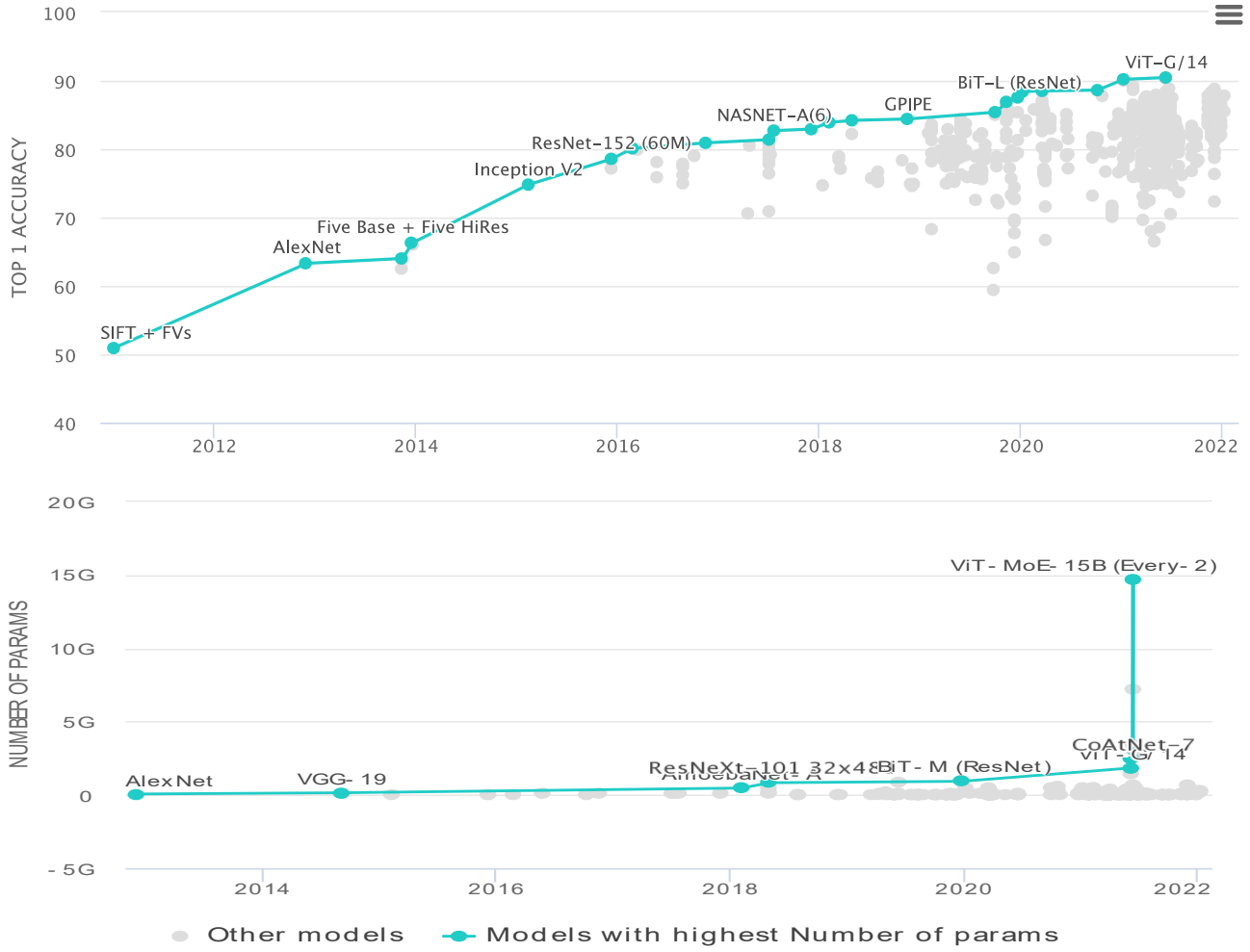
$$\min_{\boldsymbol{H}, \boldsymbol{W}, \boldsymbol{b}} \frac{1}{Kn} \sum_{k=1}^{K} \sum_{i=1}^{n} \mathcal{L}_{\mathrm{CE}}(\boldsymbol{W}\boldsymbol{h}_{k,i} + \boldsymbol{b}, \boldsymbol{y}_k) + \lambda \left\| (\boldsymbol{H}, \boldsymbol{W}, \boldsymbol{b}) \right\|_F^2$$

$$\text{where } \boldsymbol{H} := \begin{bmatrix} \boldsymbol{h}_{1,1} & \cdots & \boldsymbol{h}_{K,n} \end{bmatrix}$$

# Heavy Parameterization in CV Models

IMAGENET

- 1000 object classes
- Images:
  - 1.2 M train
  - 100K test

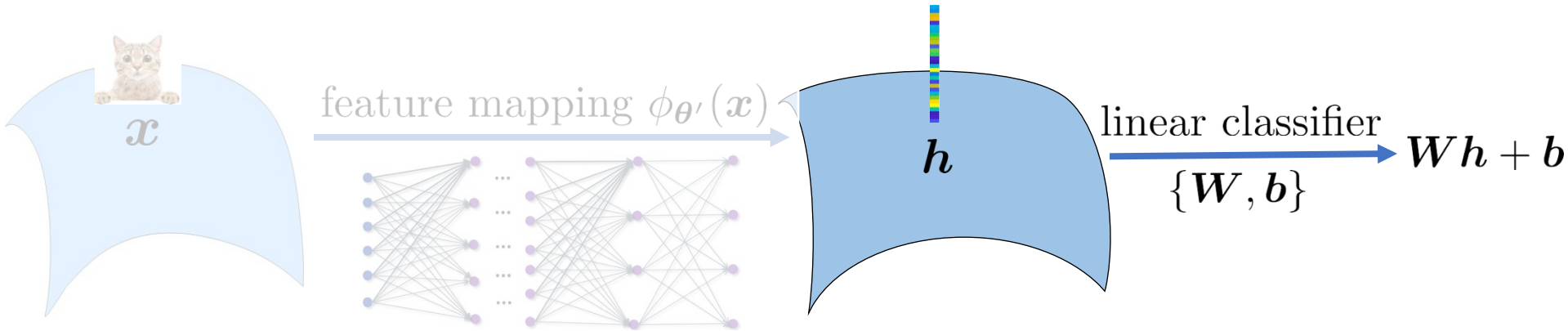



ImageNet Classification Leaderboard from paperswithcode.com

Performance & Model size keep increasing

# parameters >> # training samples

# Simplification: Unconstrained Features



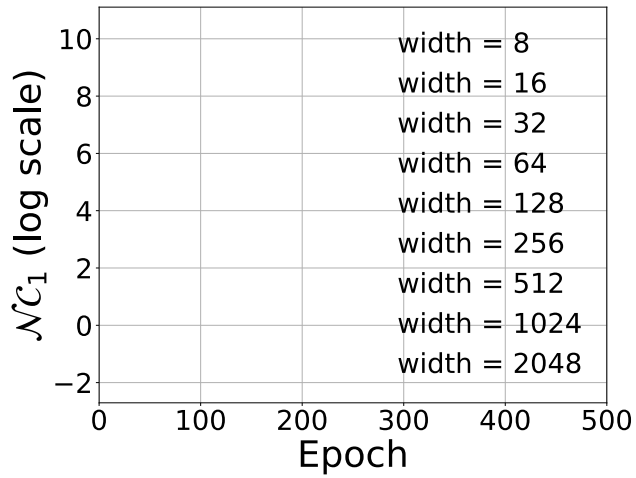

$$\min_{\boldsymbol{H},\boldsymbol{W},\boldsymbol{b}} \frac{1}{Kn} \sum_{k=1}^{K} \sum_{i=1}^{n} \mathcal{L}_{\mathrm{CE}}(\boldsymbol{W}\boldsymbol{h}_{k,i} + \boldsymbol{b}, \boldsymbol{y}_k) + \lambda \left\|(\boldsymbol{H}, \boldsymbol{W}, \boldsymbol{b})\right\|_F^2$$

- **Validity**: Modern networks are highly overparameterized, that can approximate any point in the feature space
  - also called Layer-peeled model [Fang et al'21] and has been widely used recently for studying NC

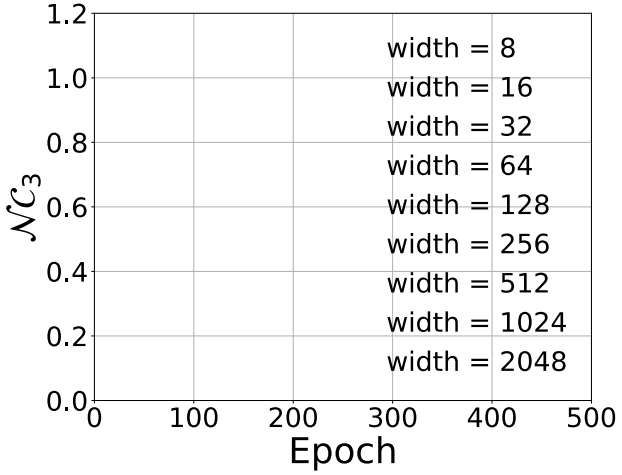

J. Lu and S. Steinerberger, Neural collapse with cross-entropy loss, ACHA, 2022
W. E and S. Wojtowytsch, On the emergence of tetrahedral symmetry in the final and penultimate layers of neural network classifiers, 2021
D. Mixon, H. Parshall, J. Pi. Neural collapse with unconstrained features, 2020
C. Fang, H. He, Q. Long, W. Su, Layer-peeled model: Toward understanding well-trained deep neural networks, 2021
W. Ji, Y. Lu, Y. Zhang, Z. Deng, W. Su, An unconstrained layer-peeled perspective on neural collapse, 2021
J. Zarka, F. Guth, S. Mallat, Separation and Concentration in Deep Networks, ICLR, 2021.
F. Graf, C. Hofer, M. Niethammer, R. Kwitt, Dissecting Supervised Contrastive Learning, ICML, 2021.
T. Ergen, M. Pilanci, Revealing the structure of deep neural networks via convex duality, ICML, 2021.
A Rangamani, M Xu, A Banburski, Q Liao, T Poggio, Dynamics and Neural Collapse in Deep Classifiers trained with the Square Loss, 2021.

# Experiment: NC Occurs on Random Labels/Inputs

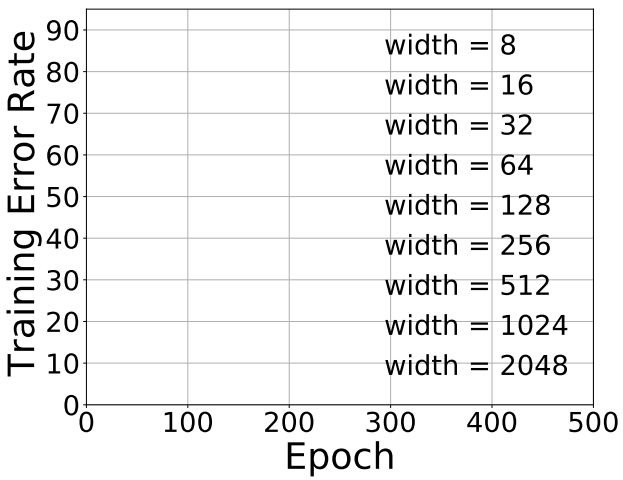
- CIFAR-10 Dataset with random labels, MLP with **varying network width**




Within-Class Variability (NC1)




Self-Duality Collapse (NC3)




Training Error

- **Validity of unconstrained features model**: Learned last-layer features and classifiers seems to be independent of input!
- The network memorizes training data in a very special way: NC
- We observe similar results on random inputs (random pixels)

# Geometric Analysis of Global Landscape

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

**Theorem (informal)** Let feature dim $d \geq$ #class $K$-1.

(**Global Optimality**) Any global solution $(\boldsymbol{W}^\star, \boldsymbol{H}^\star, \boldsymbol{b}^\star)$ must satisfy NC: $\boldsymbol{b}^\star = \boldsymbol{0}$ and

$$\underbrace{\boldsymbol{h}_{k,i}^\star = \overline{\boldsymbol{h}}_k}_{NC_1}, \quad \underbrace{\frac{\langle \overline{\boldsymbol{h}}_k^\star, \overline{\boldsymbol{h}}_{k'}^\star\rangle}{\|\overline{\boldsymbol{h}}_k^\star\|\|\overline{\boldsymbol{h}}_{k'}^\star\|} = \begin{cases} 1, & k=k' \\ -\frac{1}{K-1}, & k\neq k' \end{cases}}_{NC_2}, \quad \underbrace{\frac{\boldsymbol{w}^{k\star}}{\|\boldsymbol{w}^{k\star}\|} \to \frac{\overline{\boldsymbol{h}}_k^\star}{\|\overline{\boldsymbol{h}}_k^\star\|}}_{NC_3}$$

- $d \geq K$-1 is required to make $K$ class-mean features equal angle with cosine angle $-\frac{1}{K-1}$ (the largest possible) between each pair.
- The features have zero global mean
- If $H$ are constrained to be nonnegative (output of ReLu unit)
  - Bias term compensates for the global mean of the features $\boldsymbol{b} \to -\boldsymbol{W}\boldsymbol{h}_G$

# Geometric Analysis of Global Landscape

- [Lu et al.'22] 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.'21, Fang et al.'21, Gral et al.'21] studies constrained formulation

$$\min_{\boldsymbol{W},\boldsymbol{H}} \frac{1}{Kn} \sum_{k=1}^{K} \sum_{i=1}^{n} \mathcal{L}_{\mathrm{CE}}(\boldsymbol{W}\boldsymbol{h}_{k,i}, \boldsymbol{y}_k), \ \text{s.t.} \quad \|\boldsymbol{W}\|_F^2 \leq C_W, \|\boldsymbol{H}\|_F^2 \leq C_H$$

- These work show that any global solution has NC, but
  - What about local minima?
  - The constraint formulations are not aligned with practice

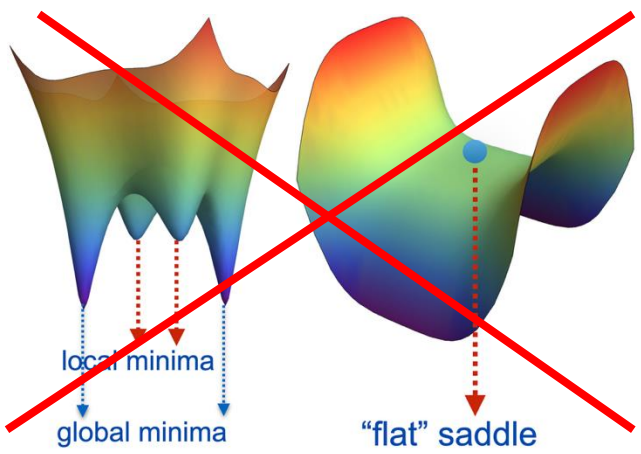

J. Lu and S. Steinerberger, Neural collapse with cross-entropy loss, ACHA, 2022
W. E and S. Wojtowytsch, On the emergence of tetrahedral symmetry in the final and penultimate layers of neural network classifiers, 2021
D. Mixon, H. Parshall, J. Pi. Neural collapse with unconstrained features, 2020
C. Fang, H. He, Q. Long, W. Su, Layer-peeled model: Toward understanding well-trained deep neural networks, 2021
W. Ji, Y. Lu, Y. Zhang, Z. Deng, W. Su, An unconstrained layer-peeled perspective on neural collapse, 2021
J. Zarka, F. Guth, S. Mallat, Separation and Concentration in Deep Networks, ICLR, 2021.
F. Graf, C. Hofer, M. Niethammer, R. Kwitt, Dissecting Supervised Contrastive Learning, ICML, 2021.
T. Ergen, M. Pilanci, Revealing the structure of deep neural networks via convex duality, ICML, 2021.
A Rangamani, M Xu, A Banburski, Q Liao, T Poggio, Dynamics and Neural Collapse in Deep Classifiers trained with the Square Loss, 2021.

# Geometric Analysis of Global Landscape

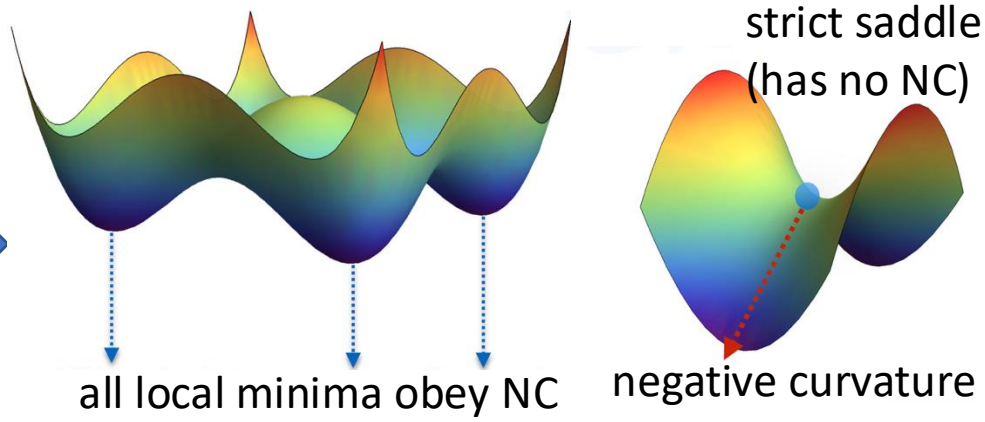
$$\min_{\boldsymbol{H},\boldsymbol{W},\boldsymbol{b}} \frac{1}{Kn} \sum_{k=1}^{K} \sum_{i=1}^{n} \mathcal{L}_{\mathrm{CE}}(\boldsymbol{W}\boldsymbol{h}_{k,i} + \boldsymbol{b}, \boldsymbol{y}_k) + \lambda \|(\boldsymbol{H}, \boldsymbol{W}, \boldsymbol{b})\|_F^2$$

**Theorem (informal)** Let feature dim $d \geq$ #class $K$-1.

(**Benign Global Landscape**) The function has no spurious local minimizer and is a strict saddle function, with negative curvature for non-global critical points.




General nonconvex problems




Our training problem

# NC always Happens

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

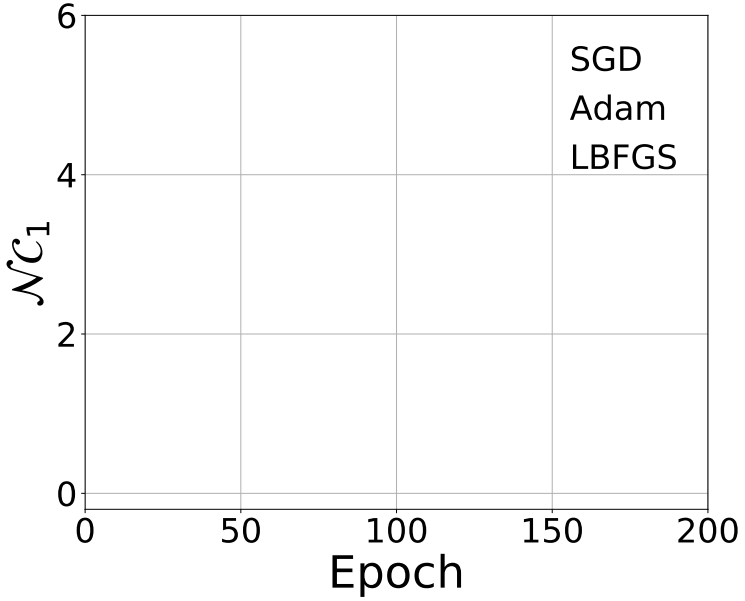
**Theorem (informal)** (**Global Optimality**) Any global solution must satisfy NC

(**Benign Global Landscape**) The loss has no spurious local minimizer and is a strict saddle function, with negative curvature for non-global critical point.

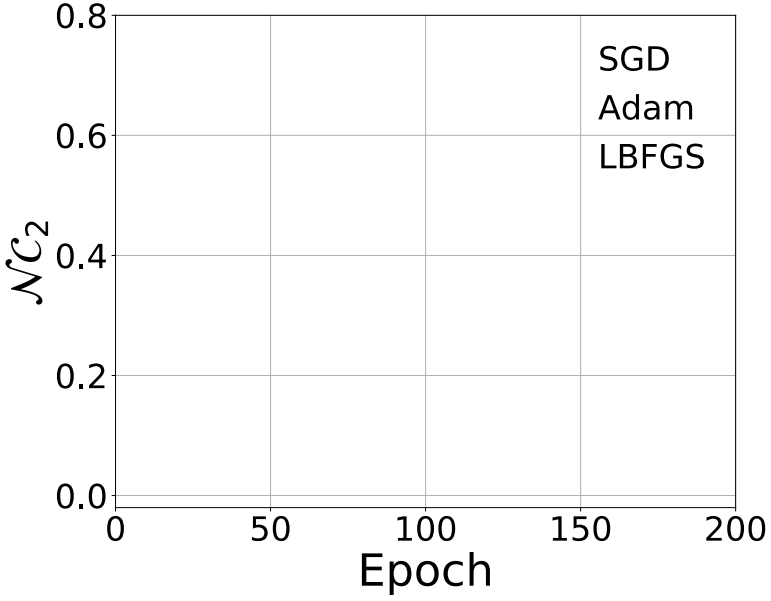
*Message: deep networks always learn Neural Collapse features and classifiers*

# Experiment: NC is Algorithm Independent

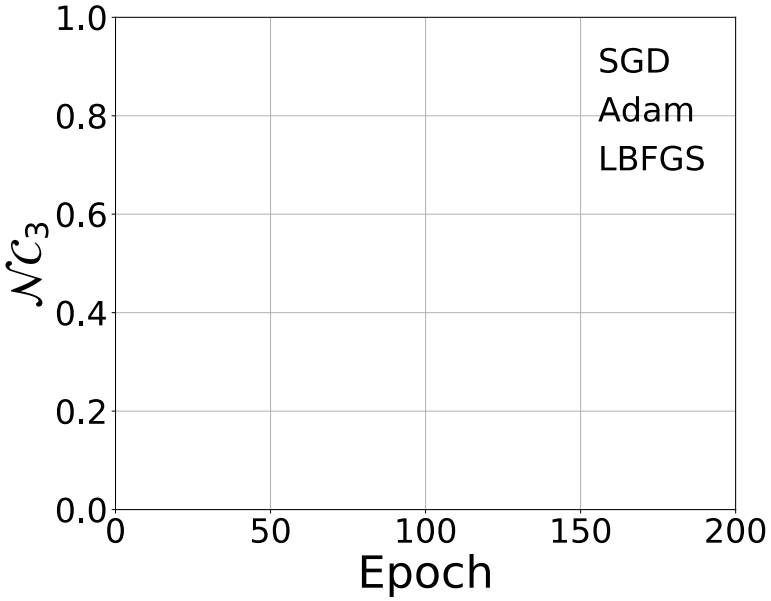
- ResNet18 on CIFAR-10 with **different training algorithms**




Within-Class Variability (NC1)




Between-Class Separation (NC2)




Self-Duality Collapse (NC3)

- The smaller the quantities, the severer NC
- NC across **different training algorithms**

# NC always Happens

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

**Theorem (informal)** (**Global Optimality**) Any global solution must satisfy NC

(**Benign Global Landscape**) The loss has no spurious local minimizer and is a strict saddle function, with negative curvature for non-global critical point.

*Message: deep networks always learn Neural Collapse features and classifiers*

- Holds for many other losses, such as focal, label smoothing, MSE
  - all losses lead to largely identical features on training data and performance on testing
- Analysis can be extended to understand other training paradigms, such as MCR$^2$, self-supervised, etc.

Zhou, You, Li, Liu, Liu, Qu, Zhu, Are All Losses Created Equal: A Neural Collapse Perspective, NeurIPS 2022.
Peng Wang, Huikang Liu, Druv Pai, Yaodong Yu, Zhihui Zhu, Qing Qu, and Yi Ma, "A Global Geometric Analysis of Maximal Coding Rate Reduction," ICML, 2024.

# All Loses Lead to Identical Features?

- We study them under the unconstrained feature model:

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

**Theorem (informal)** With feature dimension feature dim $d \geq$ #class $K$-1, the losses (CE, FL, LS, MSE, etc) lead to the same NC features and classifiers

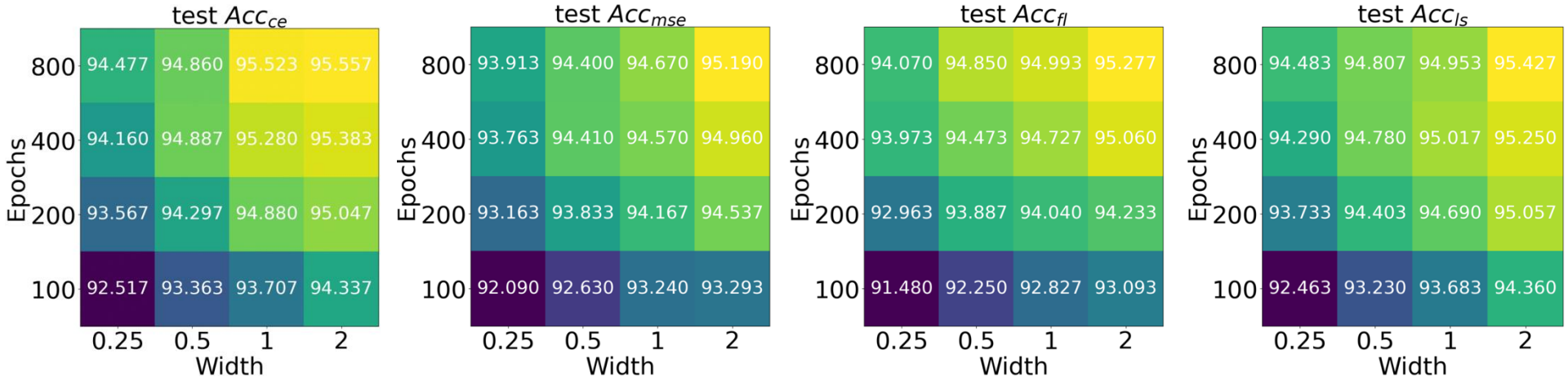
**Implication for practical network:** If network is large enough and trained longer enough

- All losses lead to largely identical features on **training data** (NC)
- All losses lead to largely identical performance on **test data**

Zhou,…, Zhu, On the Optimization Landscape of Neural Collapse Under MSE Loss: Global Optimality with Unconstrained Features, ICML 2022.
Zhou, You, Li, Liu, Liu, Qu, Zhu, Are All Losses Created Equal: A Neural Collapse Perspective, NeurIPS 2022.

# Experiments: Different Loses on CIFAR10

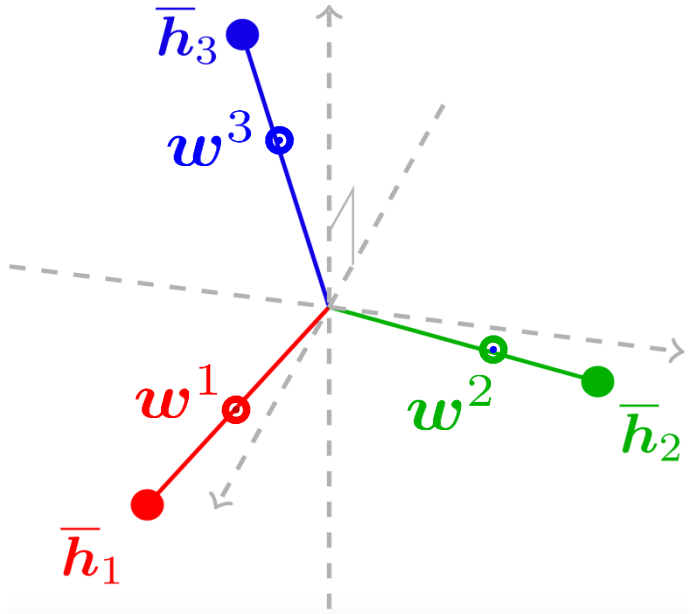
- ResNet50, CIFAR-10 with different network width and training epoches




- Left bottom corner has **larger** difference than right top corner
- If network is large enough and trained longer enough:
  - All losses lead to largely identical features on training data (NC)
  - All losses lead to largely identical performance on test data

# Exploit NC for Practical Network Training

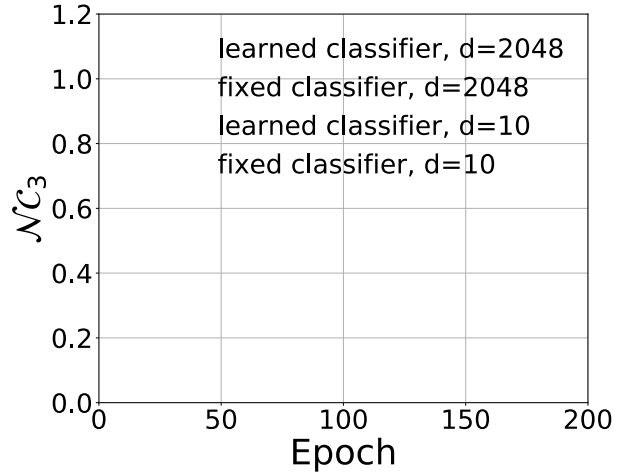
Observation: NC is prevalent, and classifier always converges to a Simplex ETF



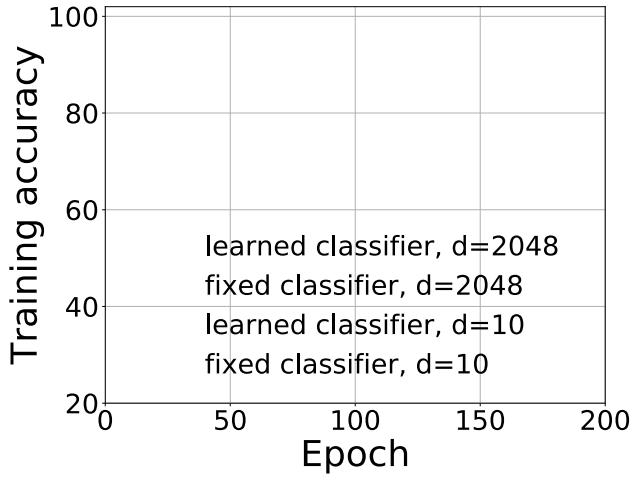

- **Implication 1: No need to learn the classifier**
  - Just fix it as a Simplex ETF
  - Save 8%, 12%, and 53% parameters for ResNet50, DenseNet169, and ShuffleNet!
- **Implication 2: No need of large feature dimension $d$**
  - Just use feature dim $d$ = #class $K$ (e.g., $d$=10 for CIFAR10)
  - Further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

# Experiment: Fixed Classifier with $d = K$

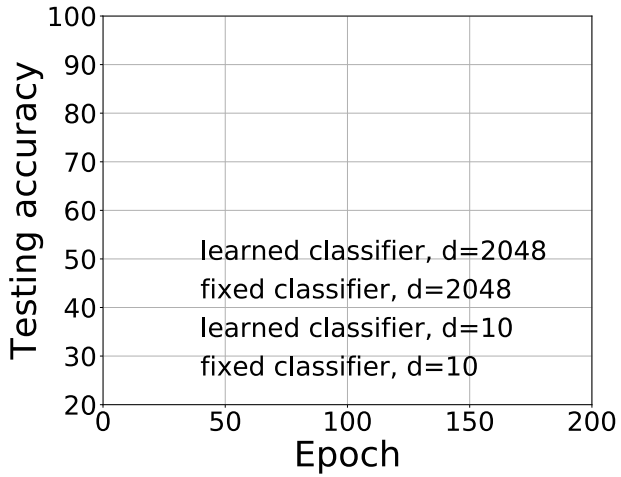
- ResNet50 on CIFAR10 with different settings
  - Learned classifier (default) **VS** fixed classifier as a simplex ETF
  - Feature dim $d$ = 2048 (default) **VS** $d$ = 10




Self-Duality Collapse (NC3)




Training Accuracy



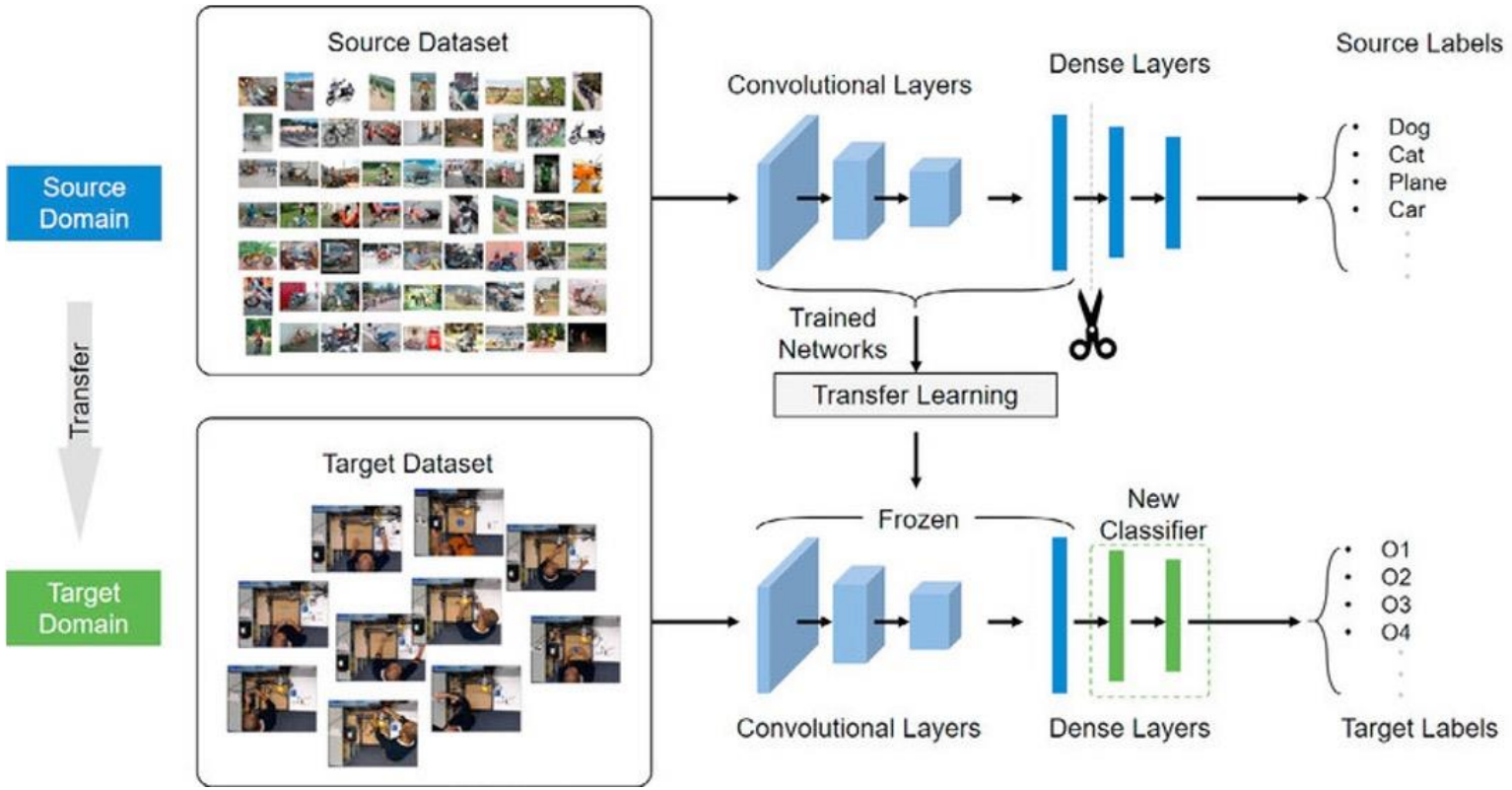

Testing Accuracy

- Training with **small** dimensional features and **fixed** classifiers achieves ***on-par performance*** with **large** dimensional features and **learned** classifiers.

# Related Work on NC

- A **non-comprehensive** overview of related work on NC
  - Theoretical analysis of NC
    - Unconstrained features model (UFM)
    - Deep unconstrained features model [Tirer & Bruna'22, Súkeník et al.'24]
    - Beyond UFM
      - global optimality [Jacot et al.'24]
      - gradient flow analysis [Min et al.'25]
    - Loss design
      - CE loss
      - MSE loss [Han et al.'22, Zhou et al.'22]
      - Supervised contrastive [Graf et al'21]
    - Multi-label learning [Li et al'24]
    - Large number of classes [Liu et al'23]
    - **Progressive NC** [Wang et al.'23]
  - Applications for understanding & improving network performance
    - Efficient training
    - **Transfer learning** [Galanti et al.'22, Li et al.'24]
    - Imbalanced learning [Fang et al.'21]
    - Continual learning [Yang et al.'23]
    - Differential privacy [Wang et al'24]
    - Robustness [Su et al'23]
    - Generalization [Hui et al'22]
    - Feature learning in intermediate layers [He & Su'23, Rangamani et al.'23]
    - LLM [Wu'24]

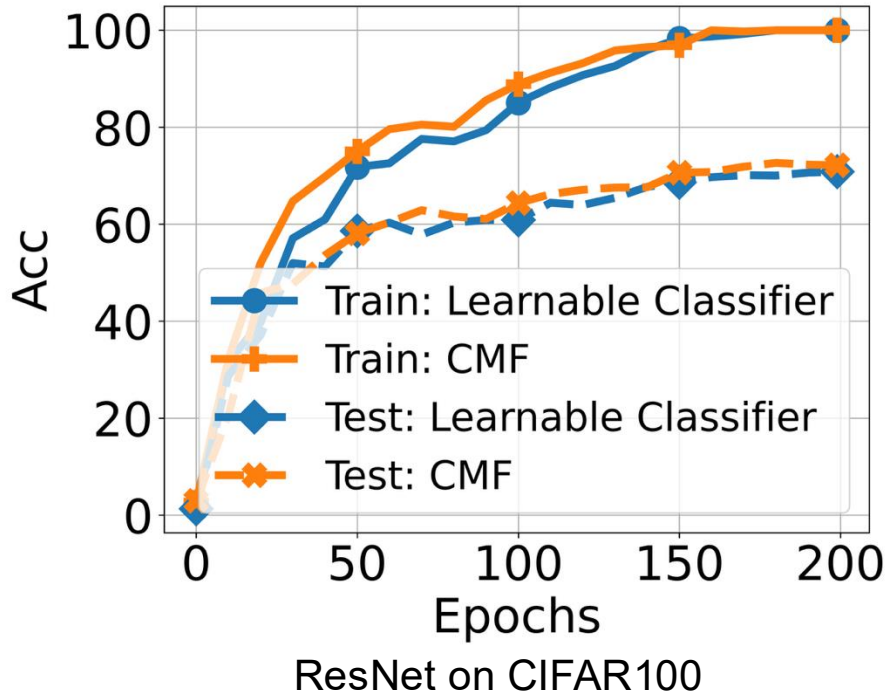
# Implications on Transfer Learning



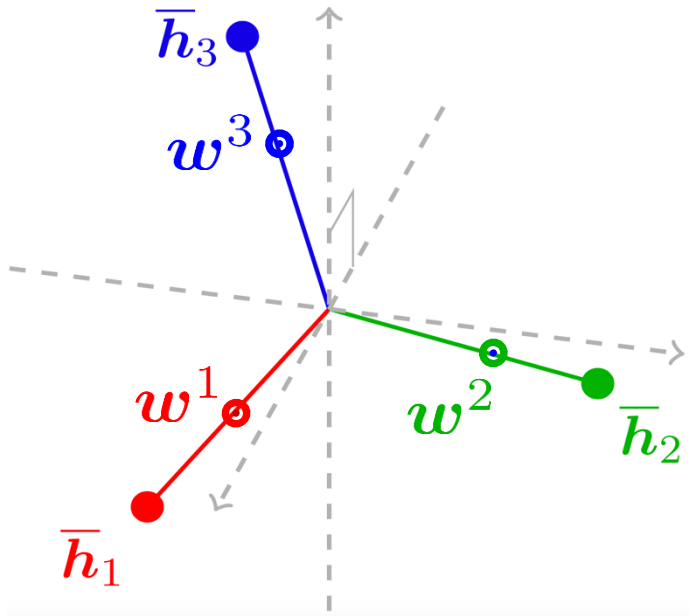

- Why transfer learning is possible
- How to improve transfer learning performance
  - initialization for the new classifier
  - efficient fine-tuning

# Experiment: Fixed Classifier as Class-mean Features

- We can also promote NC3 (self-duality) by fixing the *classifier* as the *class-mean features* (CMF) during training



ResNet on CIFAR100

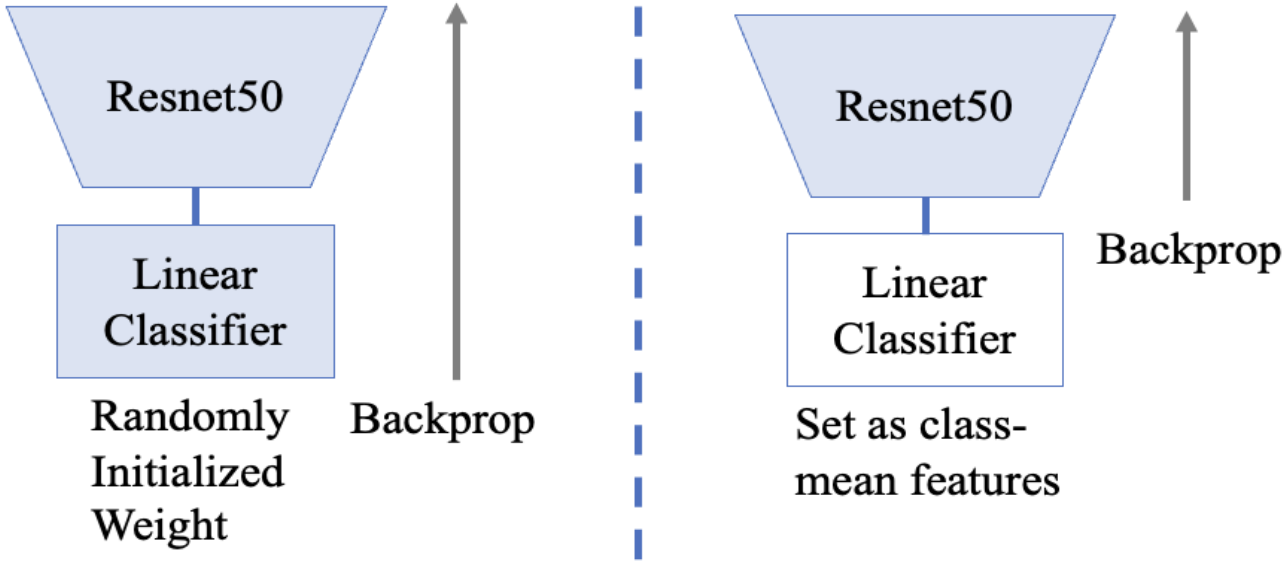



- CMF achieves **on-par performance** with learned classifiers

Jiang, Zhou, Wang, Qu, Mixon, You, Zhu, Generalized Neural Collapse for a Large Number of Classes, ICML 2024

# Experiment: Fixed Classifier as Class-mean Features

- CMF classifier improves Out-of-distribution (OOD) performance for fine-tuning
  - ResNet50 pretrained on MoCo
  - Fine-tuned for CIFAR10



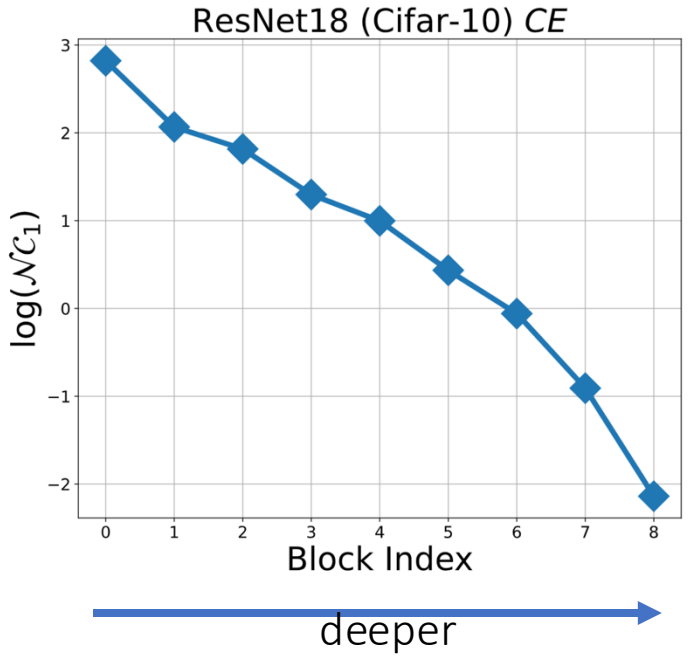

| | Learnable Classifier | CMF |
|---|---|---|
| Test on CIFAR10 (ID) | 97.00% | **98.00**% |
| Test on STL10 (OOD) | 87.42% | **90.67**% |

- Since self-duality (NC3) always hold, CMF classifier can better preserve properties of pretrained model during fine-tuning


Jiang, Zhou, Wang, Qu, Mixon, You, Zhu, Generalized Neural Collapse for a Large Number of Classes, ICML 2024
Kumar, Ananya, et al., Fine-Tuning can Distort Pretrained Features and Underperform Out-of-Distribution, ICLR 2022.

# Progressive separation from shallow to deep layers

- From shallow to deep layers (effect of depths): progressive compression and separation

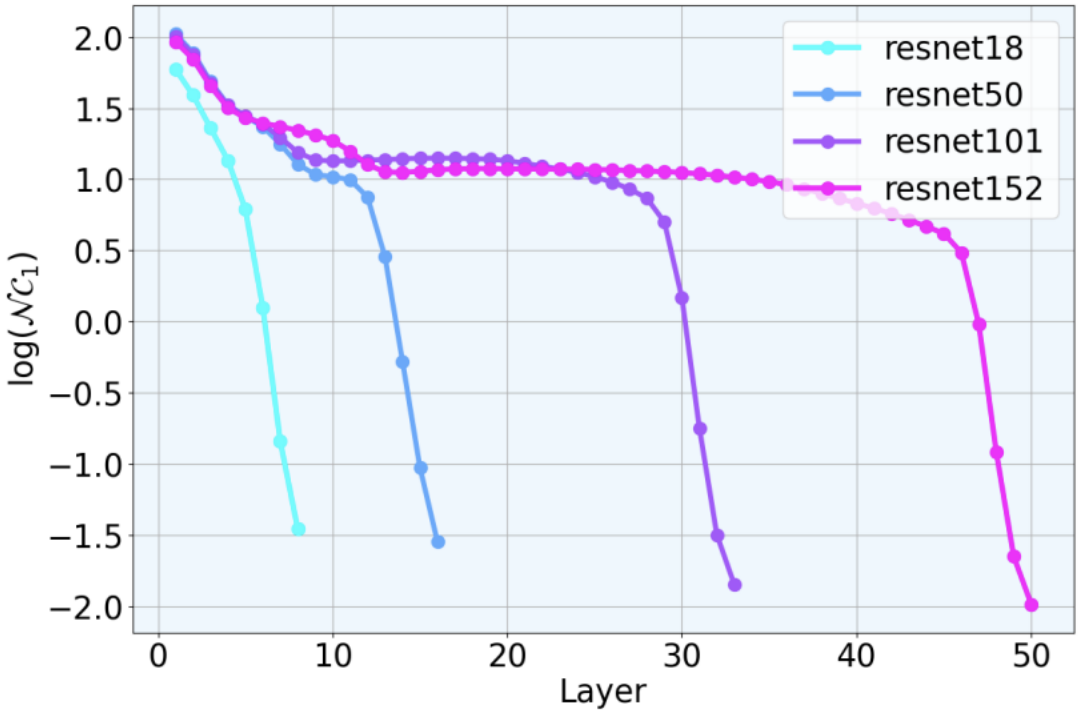




- Effect of depth: creates progressive separation and collapse

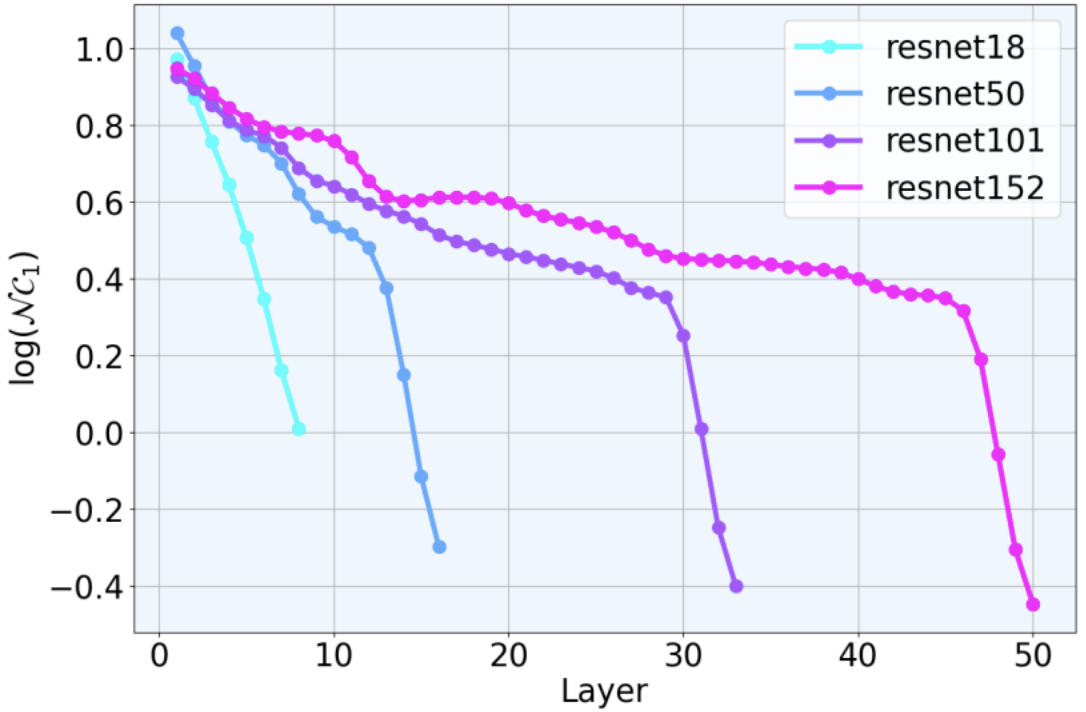
He, Hangfeng, and Weijie J. Su. "A law of data separation in deep learning," PNAS 2023.
Wang, Li, Yaras, Hu, Zhu, , Balzano, Qu, Understanding Deep Representation Learning via Layerwise Feature Compression and Discrimination, JMLR 2025
Jinxin Zhou, Jiachen Jiang, Zhihui Zhu, Are all layers created equal: A neural collapse perspective, CPAL 2025.

# Are all Layers Created Equal?

- Scaling up the model makes middle layers redundant
- Deeper layers excel at enhancing neural collapse (prone to memorization); other layers are good for transfer learning




CIFAR10



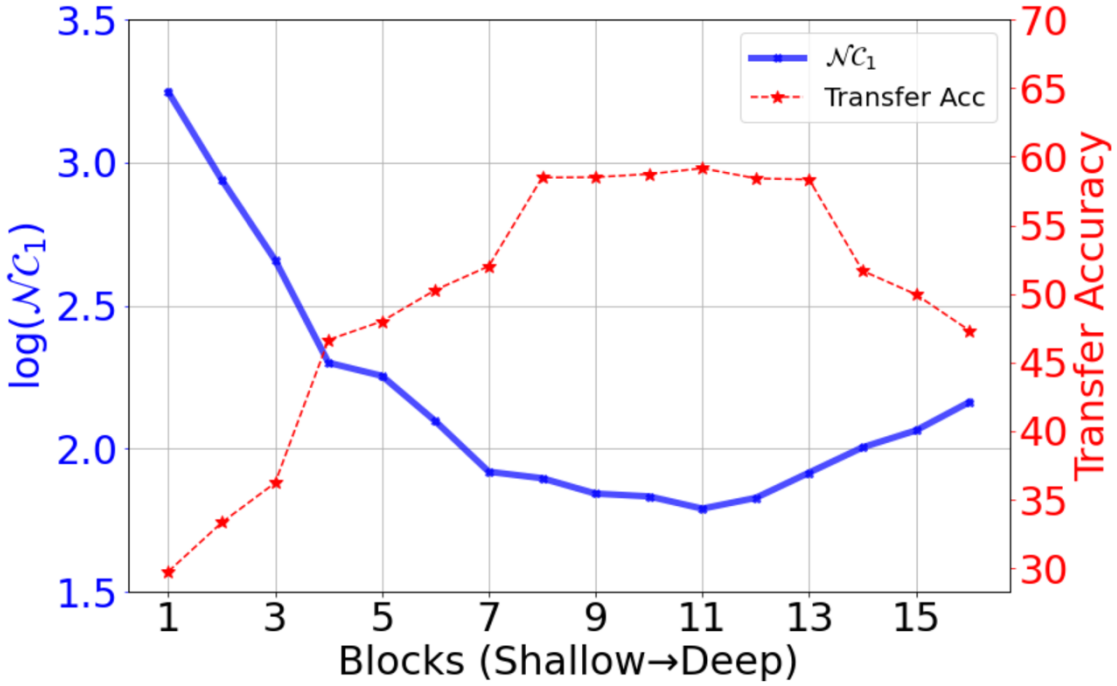

ImageNet

Jinxin Zhou, Jiachen Jiang, Zhihui Zhu, Are all layers created equal: A neural collapse perspective, CPAL 2025.

# Progressive Separation is Transferable

- Progressive separation is transferable among different tasks (common in medical domain)
  - ResNet-34 pre-trained on ImageNet
  - Evaluate on CIFAR10
  - Model is fixed without fine-tuning
  - Train a linear classifier on top of the features
- Layer-wise NC exhibits two phases on downstream tasks:
  - Phase 1: progressively decreasing (universal feature mapping)
  - Phase 2: progressively increasing (specific feature mapping) fine-tune this layer!

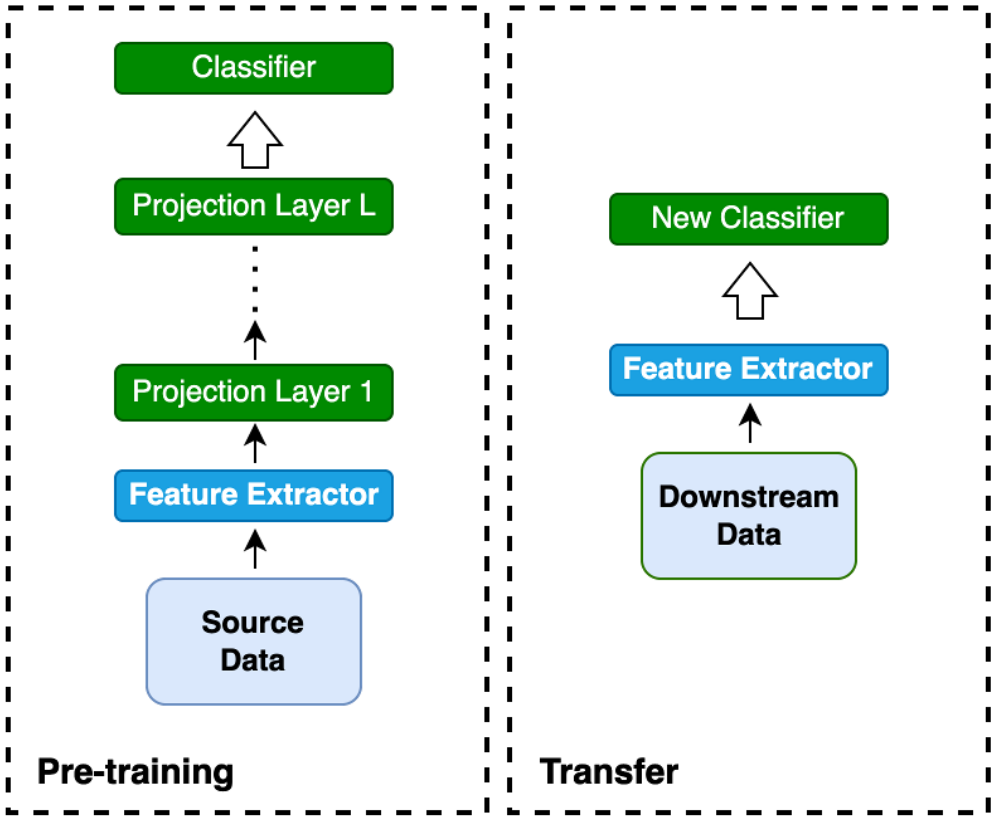
ImageNet pretrained model evaluated on CIFAR10 (no fine-tuning)



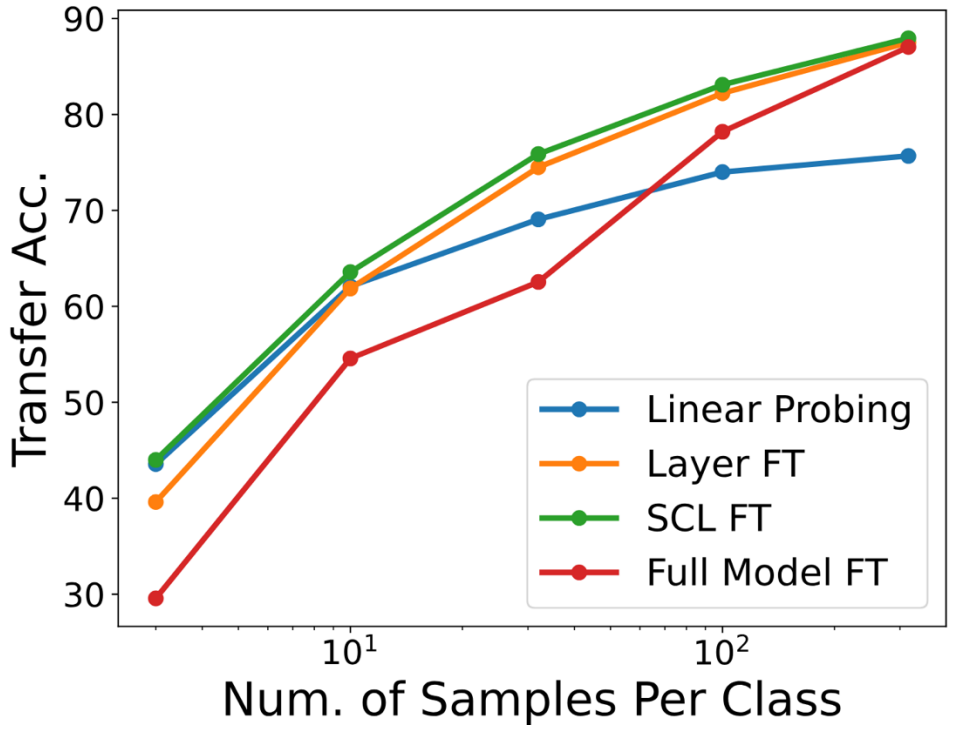

Xiao Li, Sheng Liu, Jinxin Zhou, Xinyu Lu, Carlos Fernandez-Granda, Zhihui Zhu, Qing Qu, Principled and Efficient Transfer Learning of Deep Models via Neural Collapse, TMLR 2024.

# Efficient Layer Fine-tuning

Fine-tuning one key intermediate layer is sufficient



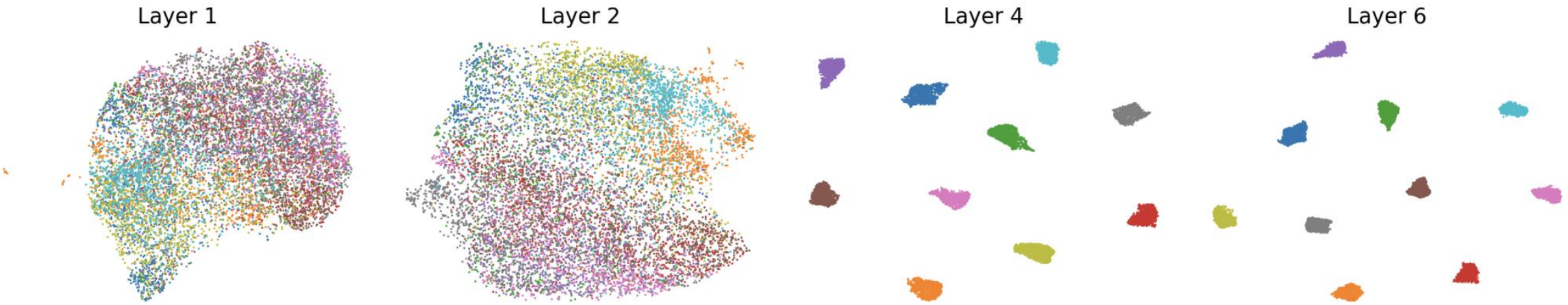

(a) Illustration of layer fine-tuning




(b) Fine-tuning results on CIFAR-10

**The objective of learning**:

Transform nonlinear and complex data to
a linear, compact, and structured representation.

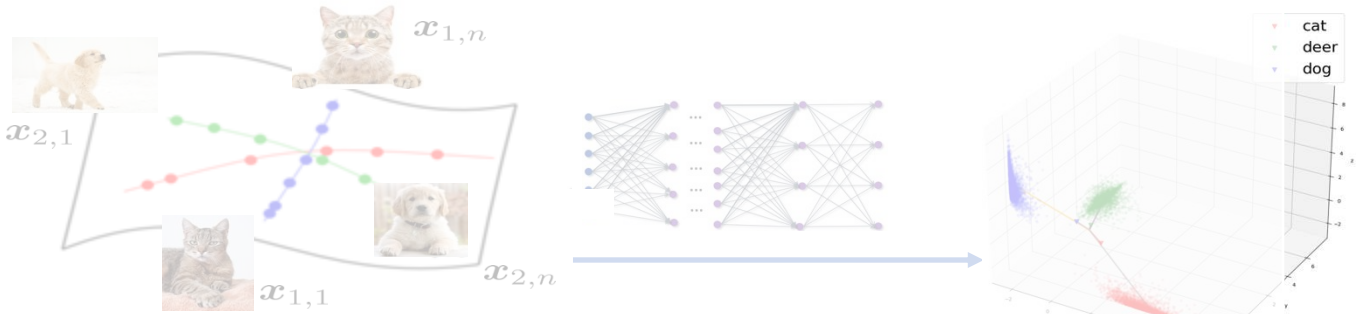



**The effect of depth**:

Progressively achieve this goal through layer-wise transformation

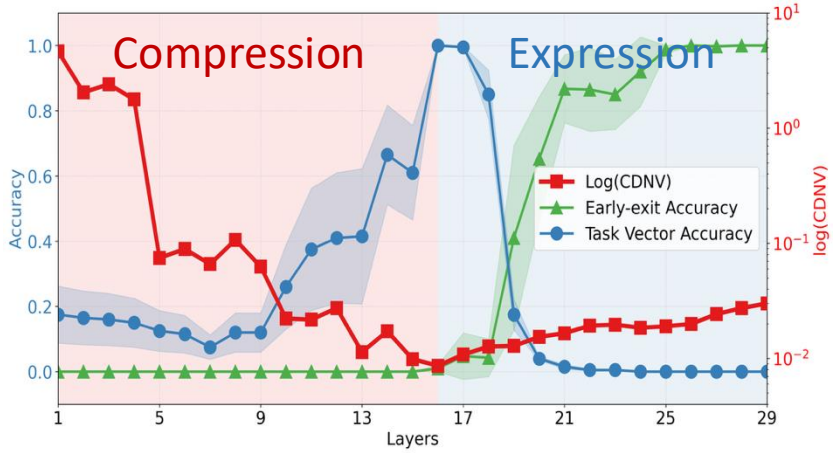
# Outline

- Low-D representation in Deep Classifiers:
  - Neural Collapse (NC) Phenomena
  - Understanding NC from optimization
  - Progressive NC and transferability



Data in the input space

- Low-D representation in Generative Models (LLM)
  - Compression-to-Expression Phenomena



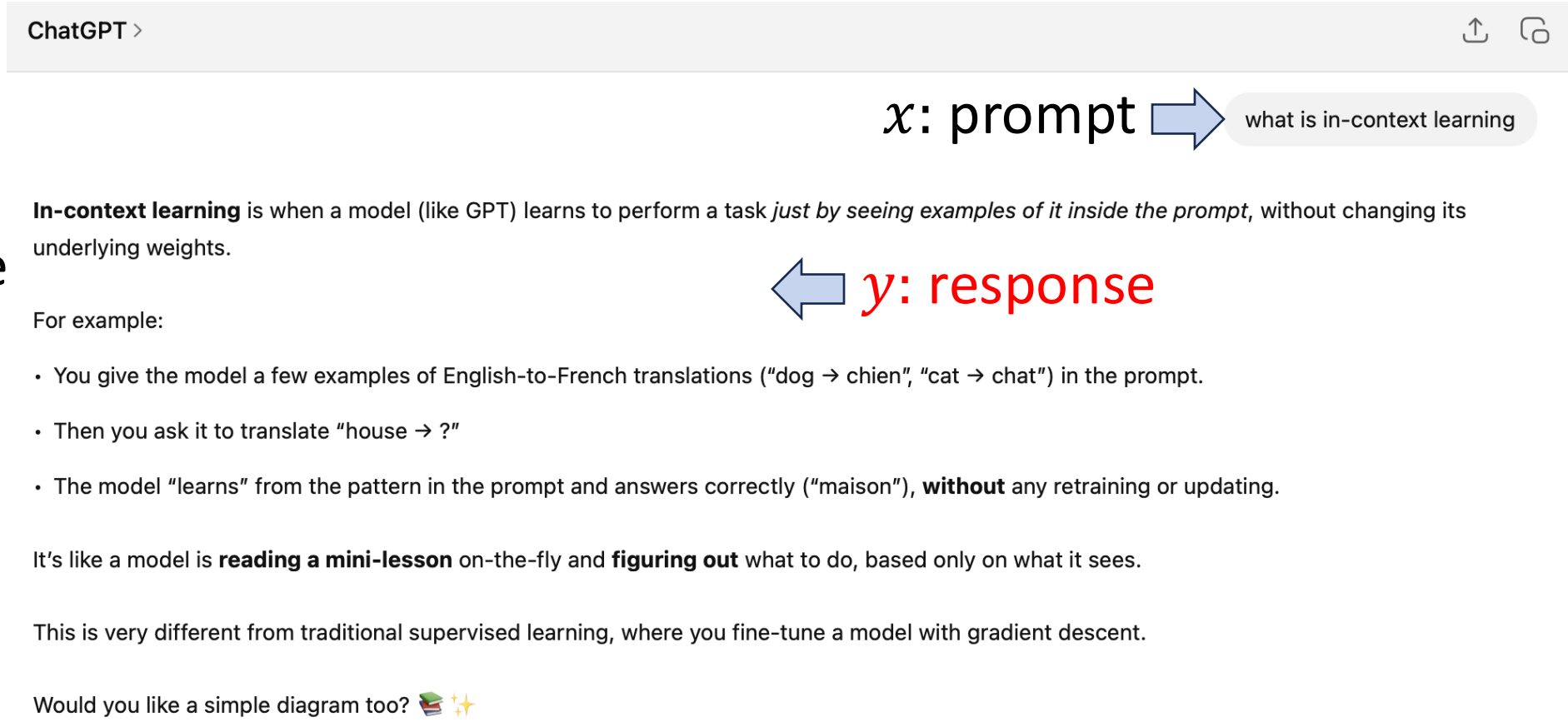

- Conclusion

# Generative Models

Two popular and effective approaches to learn distribution $P(x_1, x_2, \ldots, x_n)$

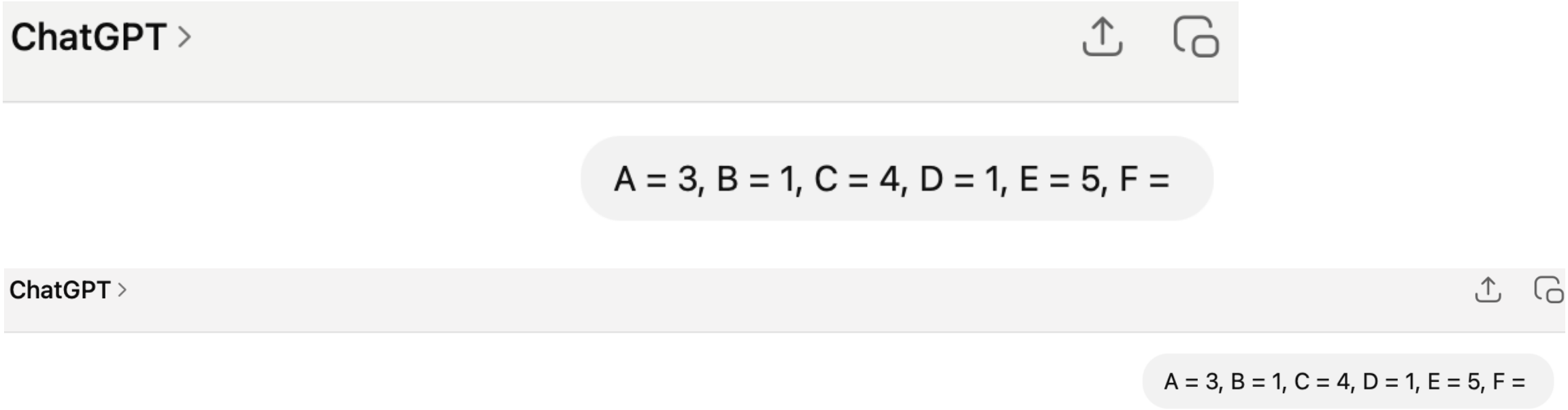
- diffusion model (see Session III by Liyue and Qing, afternoon)
- autoregressive model: $P(x_1)P(x_2|x_1) \ldots P(x_n|x_1, \ldots, x_{n-1})$

- Natural language conversation $P(y|x)$

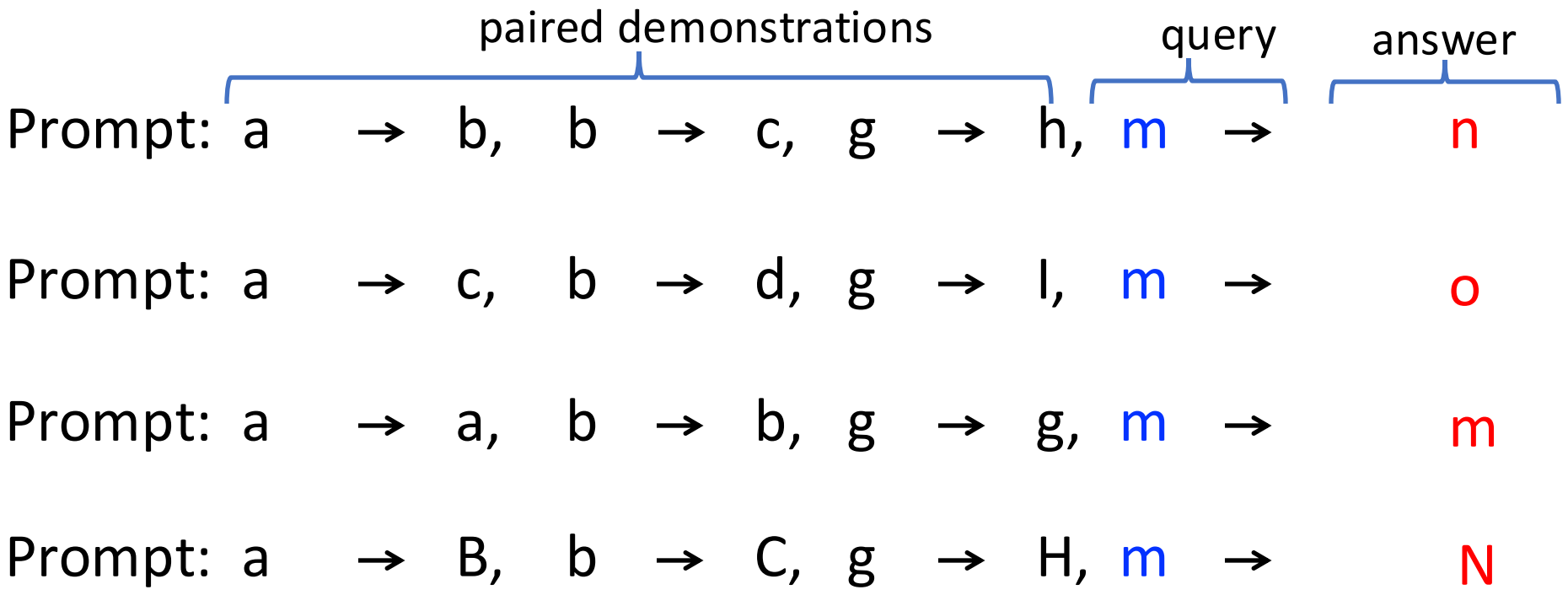


# In-context Learning

- LLM can “learn” to solve tasks without weight updates

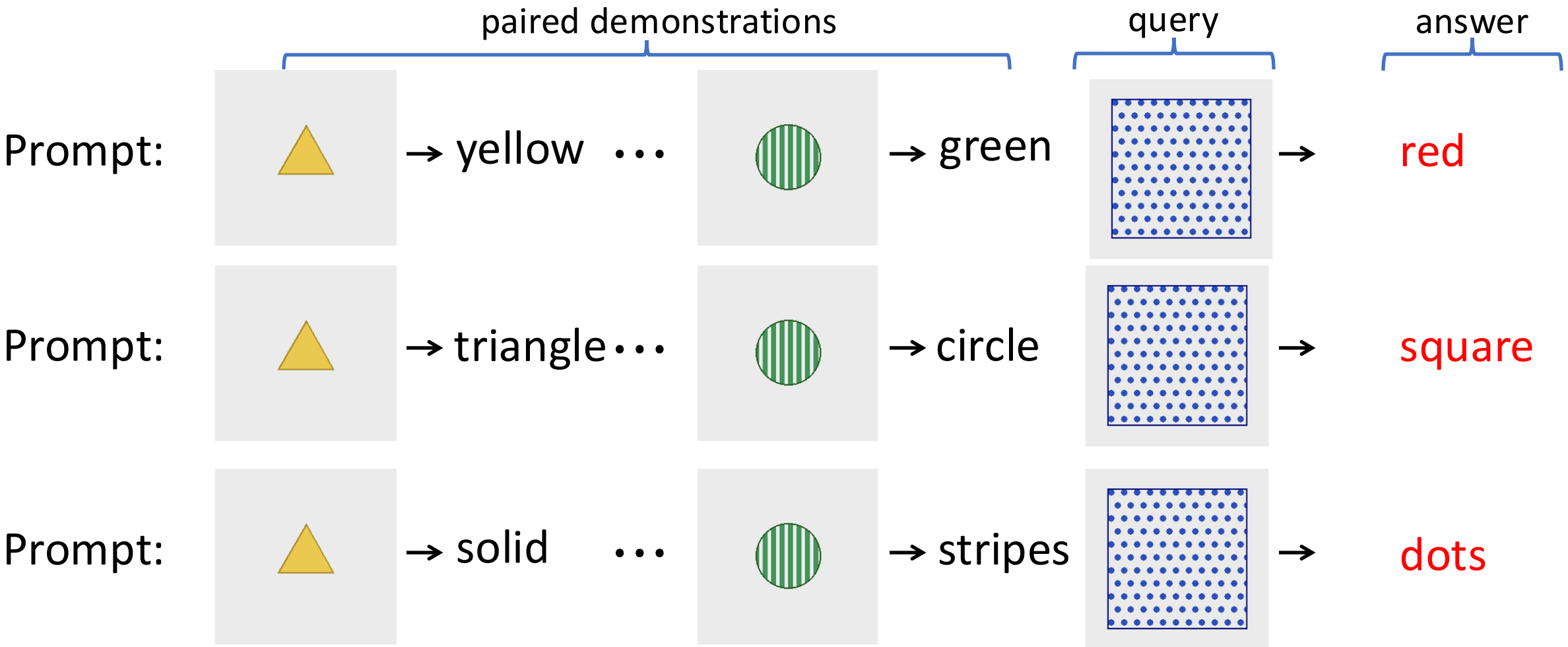


# Letter Prediction Tasks




- Step 1: identify the “rule/task/pattern” from the demonstrations
- Step 2: apply it to the query token “m”
- Does the LLM perform ICL in the same way?

# In-context Learning




- Step 1: identify the “rule/task/pattern” from the demonstrations
- Step 2: apply it to the query token
- Does the MLLM perform ICL in the same way?

# How Do LLM Perform ICL

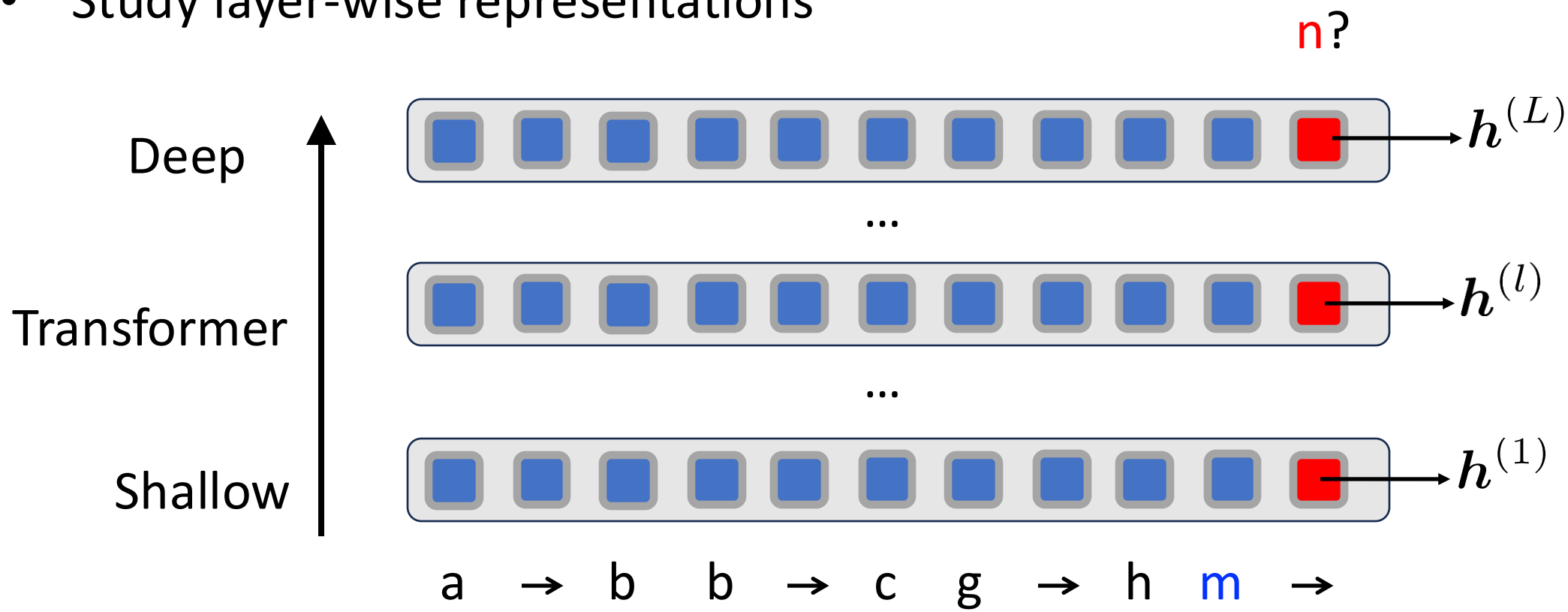
- The models are only pre-trained to predict next token. Why can they learn the input-output relationship in the prompt?

- Different approaches for understanding the ICL ability
  - Bayesian Perspective [Xie et al., 2021]
  - Meta-learning Perspective [Chen et al., 2021]
  - Meta-optimizer perspective [Oswald et al 2022, Ahn et al, 2023]
  - Mechanistic Interpretations (induction heads in copy problems [Olsson et al., 2022])
  - etc.
- But less emphasis on the internal representations within models

*How do MLLMs extract and differentiate task information from shallow to deep layers during in-context learning?*

# Tracing Hidden Representation

- Study layer-wise representations




- Denote the hidden representation of the last token at layer $l$ as $h^{(l)}$, knowns as task vector or in-context vector


Hendel, Geva, Globerson, In-Context Learning Creates Task Vectors, 2023.

S Liu, H Ye, L Xing, J Zou, In-context vectors: Making in context learning more effective and controllable through latent space steering, ICML 2024.

# Metric for Compression & Discrimination

- For each ICL task $t$ (e.g. next letter prediction), generate $N$ promots

Prompt 1: $a \to b, b \to c, c \to d, d \to$ $\boldsymbol{h}_{1,t}^{(l)}$

Prompt 2: $d \to e, l \to m, s \to t, u \to$ $\boldsymbol{h}_{2,t}^{(l)}$

…

Prompt $N$: $m \to n, f \to g, i \to j, k \to$ $\boldsymbol{h}_{N,t}^{(l)}$

- Define mean vector and variance

$$\bar{\boldsymbol{h}}_t^{(\ell)} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{h}_{i,t}^{(\ell)} \qquad \mathrm{var}_t^{(\ell)} = \frac{1}{N}\sum_{i=1}^{N} \left\| \boldsymbol{h}_{i,t}^{(\ell)} - \bar{\boldsymbol{h}}_t^{(\ell)} \right\|_2^2$$

# Metric for Compression & Discrimination

- **TDNV** (**T**ask-**D**istance **N**ormalized **V**ariance) to capture the level of compression and separation

$$\mathrm{TDNV}^{(\ell)} = \sum_{t \neq t'} \frac{\mathrm{var}_t^{(\ell)} + \mathrm{var}_{t'}^{(\ell)}}{\left\| \bar{\boldsymbol{h}}_t^{(\ell)} - \bar{\boldsymbol{h}}_{t'}^{(\ell)} \right\|_2^2}$$

- Within-task Variance: how well the representation from the same task are compressed towards its own task mean
- Between-task Distance: how well the task means of different task are discriminated from each other

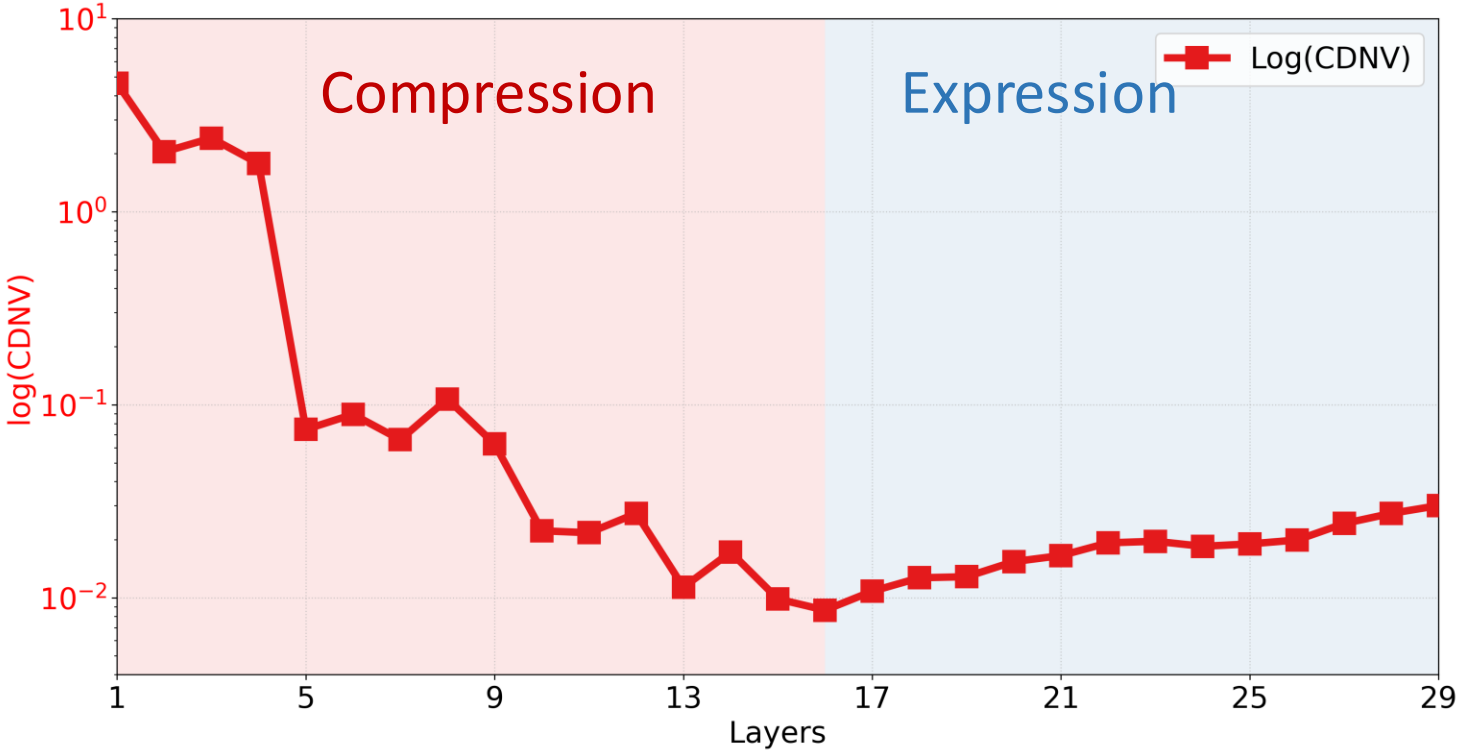
Following a similar conceptual framework to Class-Distance Normalized Variance (CDNV)[6] by viewing each task as a class.

# From Compression to Expression/Expansion

Progressive compression and then expression from shallow layers to deep layers

| Task Groups | Task | Example |
|---|---|---|
| Letter-to-Letter | Copy Letter | a → a |
| | Next Letter | a → b |
| | To Uppercase | a → A |
| | Prev Letter | b → a |
| | Next 2 Letter | a → c |

Algorithmic Tasks

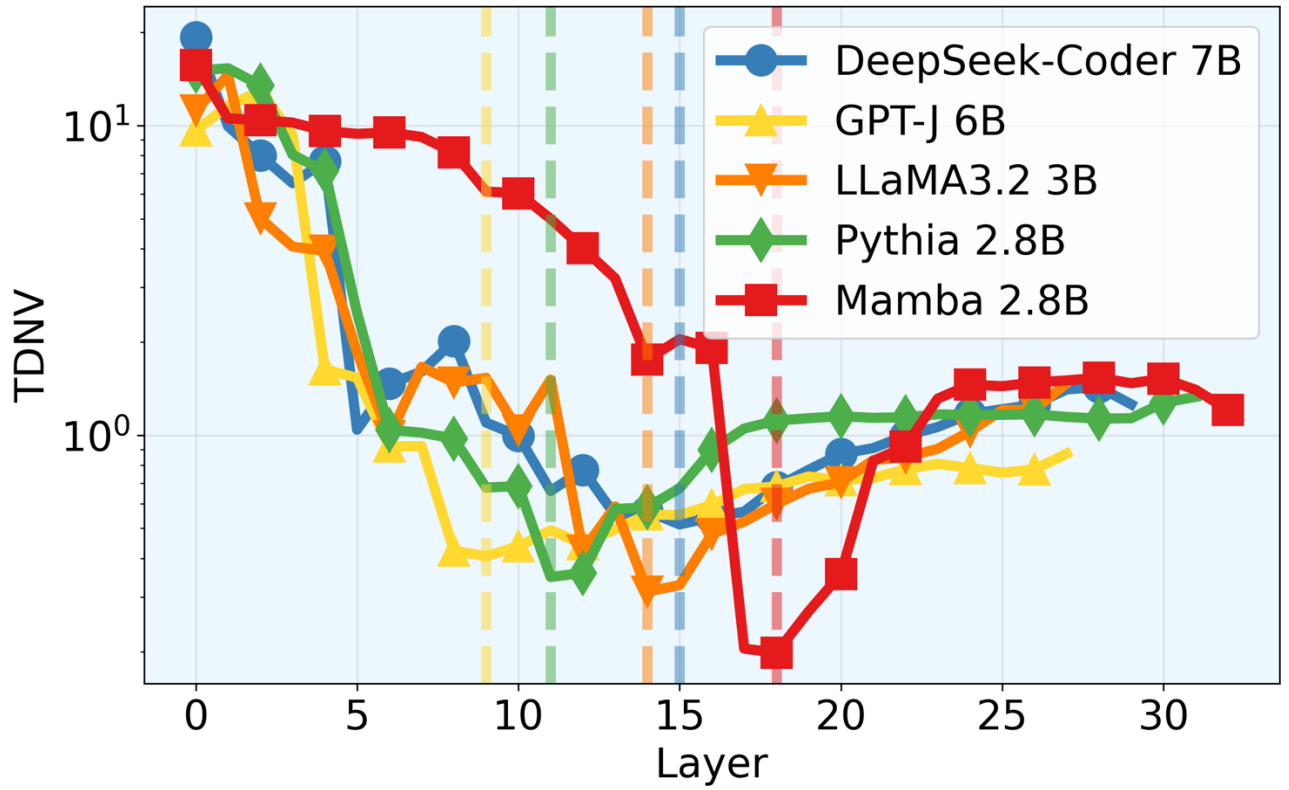



- Compression phase: the model produces compact & discriminative representation that captures the "rule/task/pattern" from the input data
- Expression phase: apply the compact representation to the input query

Jiachen Jiang, Yuxin Dong, Jinxin Zhou, Zhihui Zhu, From Compression to Expression: A Layerwise Analysis of In-Context Learning, 2025

# Prevalence of the Phenomenon

Universality across model architectures (Transformer & State-Space Model)

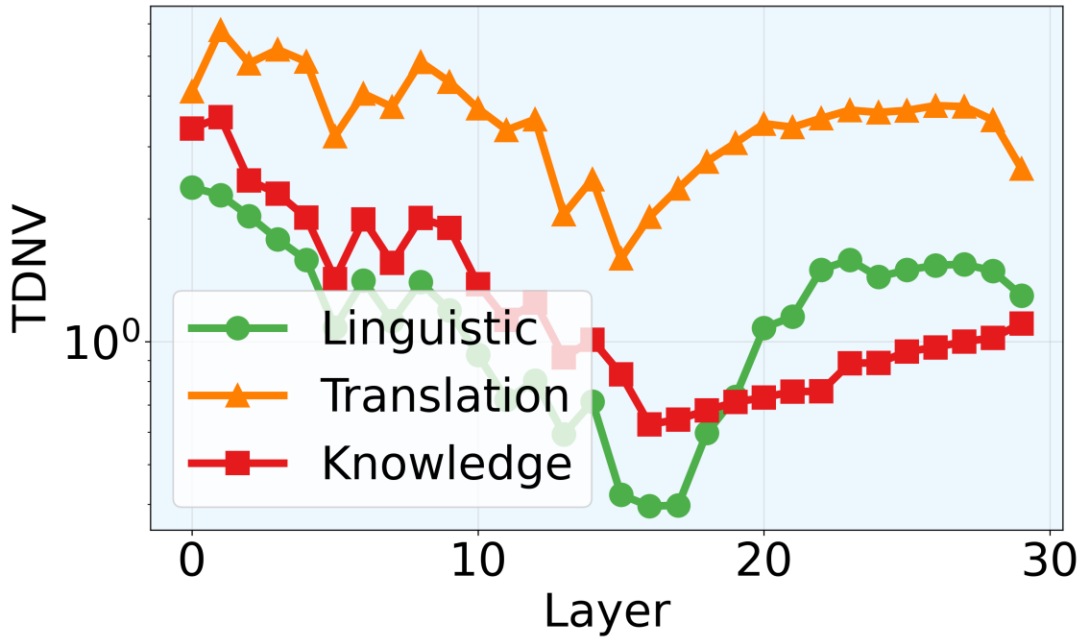


# Prevalence of the Phenomenon

## Universality across task domains

### Symbolic ICL (Beyond Algorithmic)

| | | |
|---|---|---|
| Translation | French → English | bonjour → hello |
| | Spanish → English | gracias → thank you |
| | English → French | goodbye → au revoir |
| | English → Italian | music → musica |
| | English → Spanish | thank you → gracias |
| Linguistic | Antonyms | hot → cold |
| | Plural → Singular | cats → cat |
| | Present Simple → Gerund | run → running |
| | Present Simple → Past Perfect | walk → had walked |
| | Present Simple → Past Simple | jump → jumped |
| | Singular → Plural | dog → dogs |
| Knowledge | Country → Capital | France → Paris |
| | Football Player → Position | Lionel Messi → Forward |
| | Location → Continent | Brazil → South America |
| | Location → Country | Kyoto → Japan |
| | Location → Language | Egypt → Arabic |
| | Location → Religion | India → Hinduism |

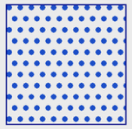



Hendel, Geva, Globerson, In-Context Learning Creates Task Vectors, 2023.

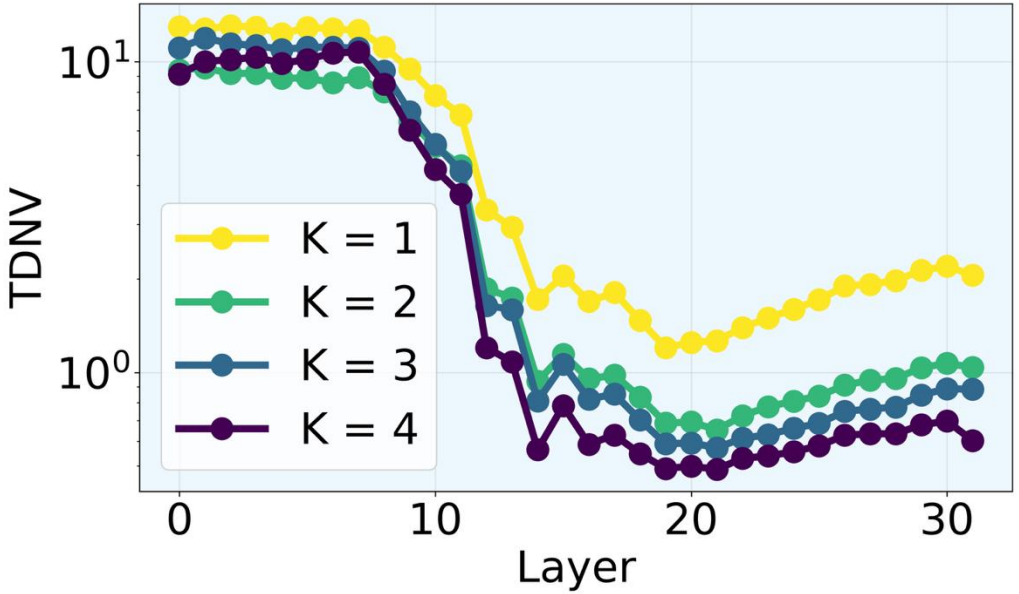
# Prevalence of the Phenomenon

## Universality across task domains

### Multimodality ICL (Beyond Language)

| image | color | shape | size | texture |
|---|---|---|---|---|
|  | yellow | triangle | small | solid |
|  | blue | square | large | dots |
|  | red | star | large | solid |
|  | black | pentagon | large | checker |
|  | green | circle | small | stripes |

| Attribute | Labels |
|---|---|
| Color | red, green, blue, yellow, black |
| Shape | circle, square, triangle, pentagon, star |
| Size | small, medium, large |
| Texture | solid, stripes, dots, checker |

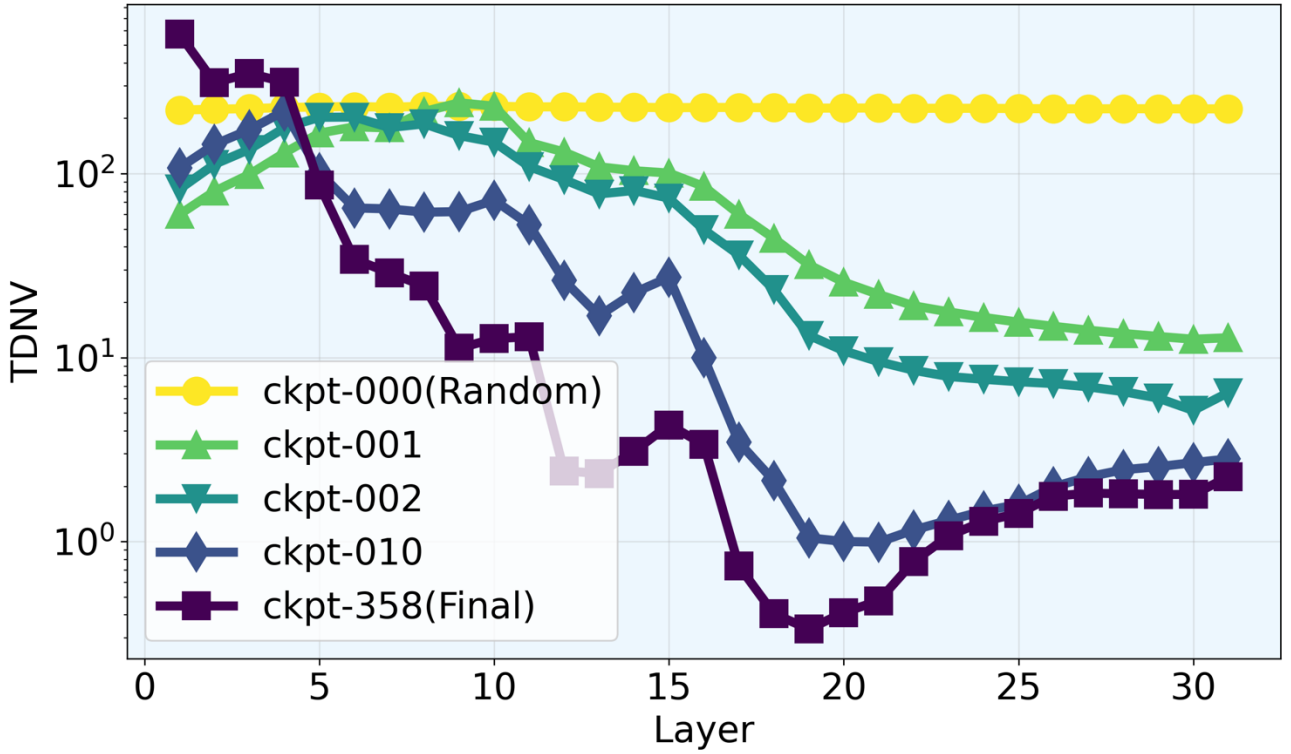



Model: Qwen-VL

Brandon Huang, Chancharik Mitra, Assaf Arbelle, Leonid Karlinsky, Trevor Darrell, Roei Herzig, Multimodal Task Vectors Enable Many-Shot Multimodal In-Context Learning, NeurIPS 2024

# Compression-Expression Emerges During Training

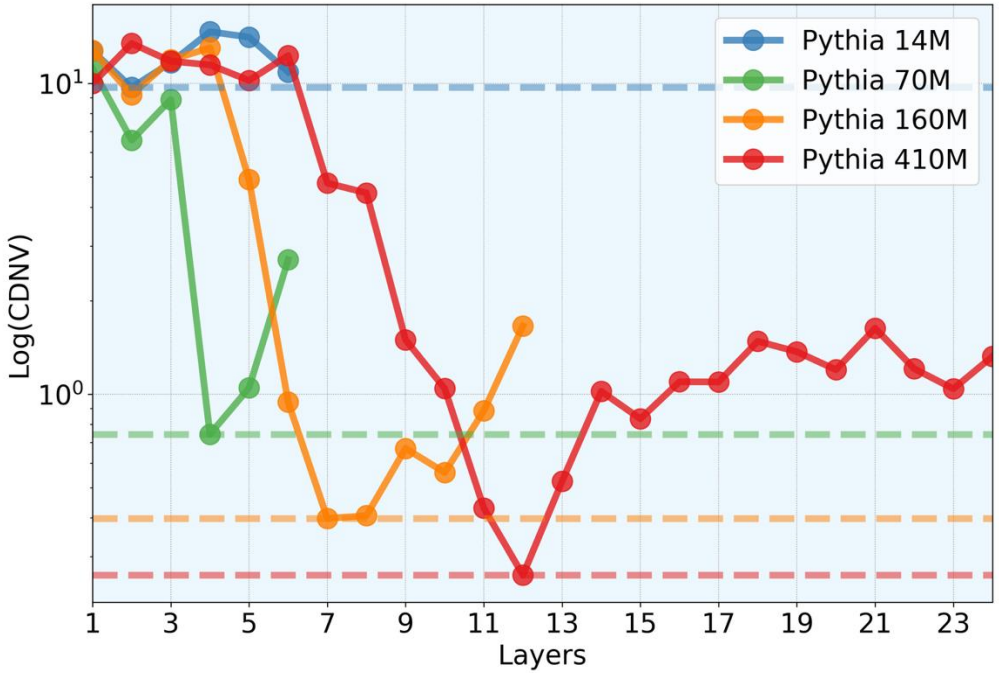
Emergence during training

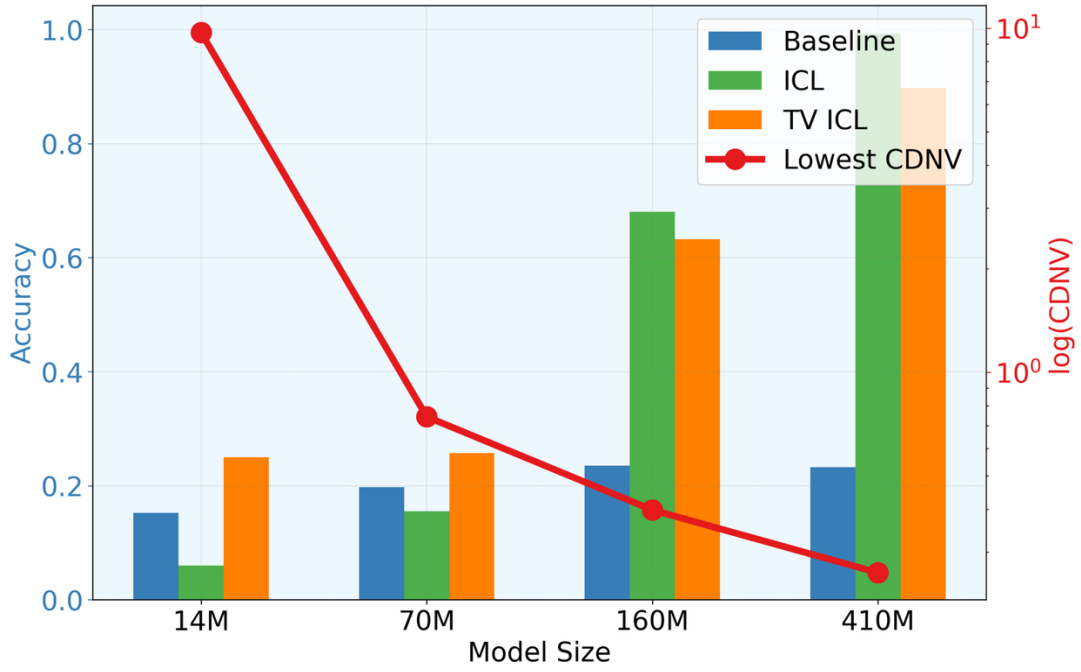



- Before training: flat TDNV values across all layers
- During training: distinct U-shape curve emerges and deepens

# Scaling Model Promotes Compression

Scaling up model size leads to more compressed representations

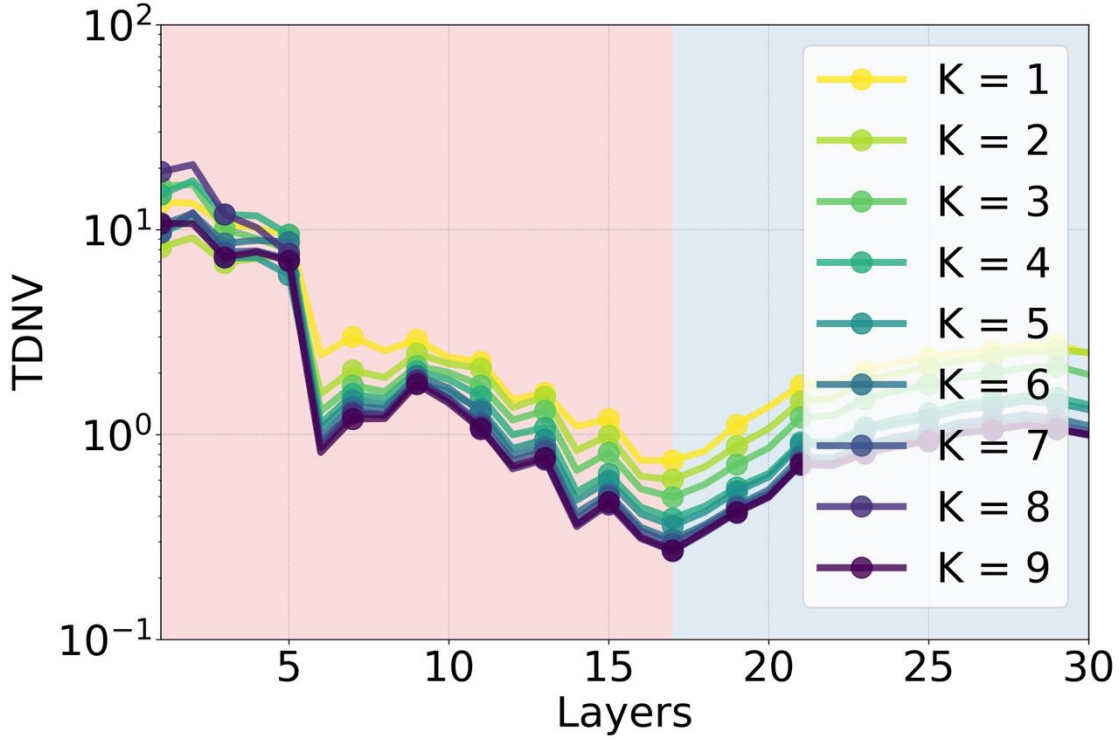



Compressed level correlates with the ICL performance.

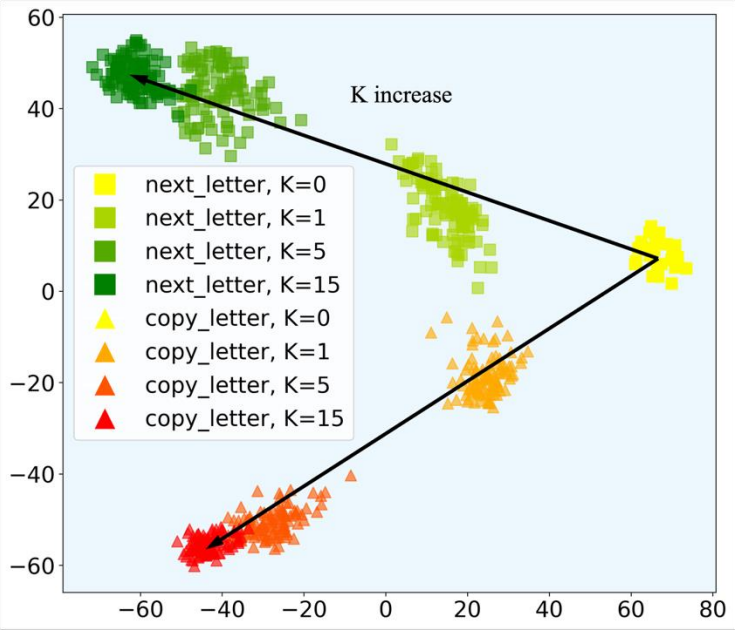
# Number of Demonstrations Promotes Compression

Increasing in-context lengths (num of demonstrations $K$) leads to more compressed representations.



# Bias-variance Decomposition

- Different tasks induce task vectors in distinct directions, yet each task follows a consistent direction
- The variance within each task decreases

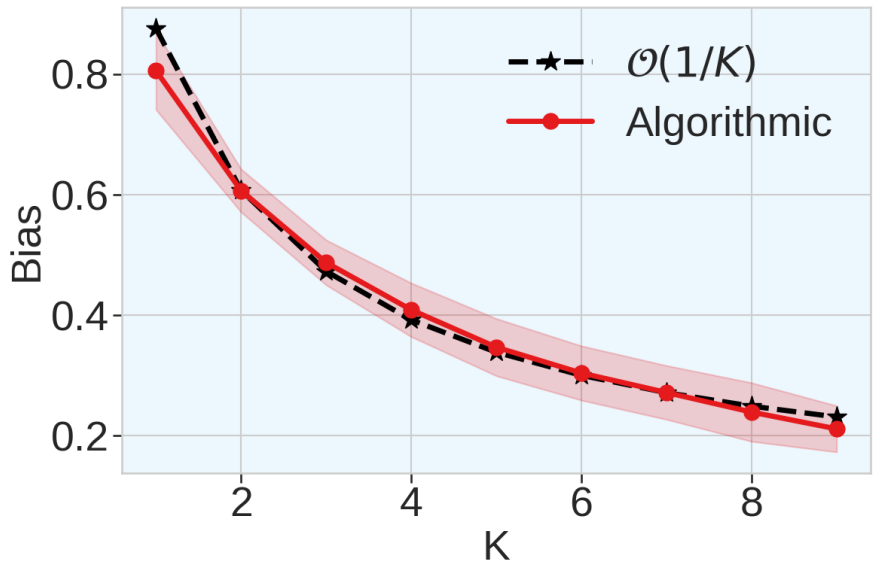


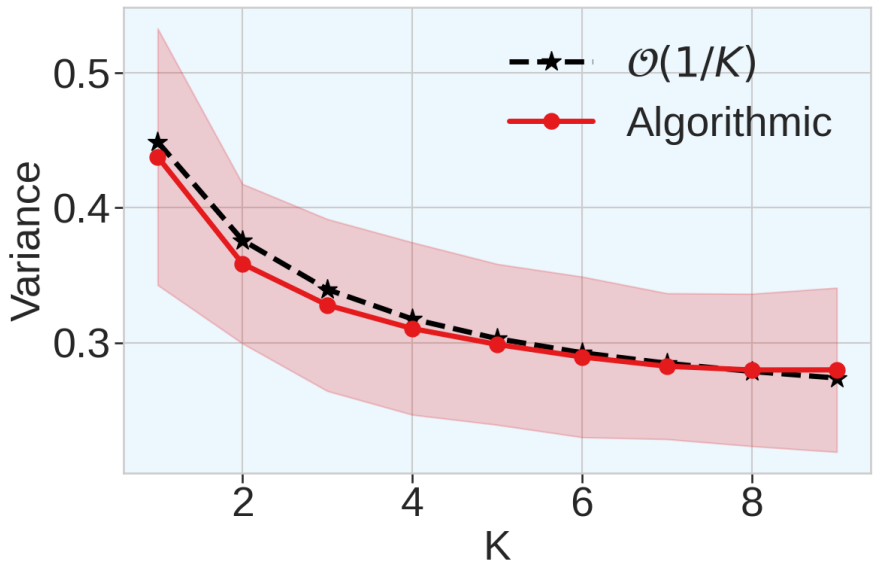

$$\boldsymbol{h}_{i,t}(K) = \underbrace{\boldsymbol{\mu}_t(\infty)}_{\text{ideal representation}} + \underbrace{\boldsymbol{\mu}_t(K) - \boldsymbol{\mu}_t(\infty)}_{\text{bias}} + \underbrace{\boldsymbol{h}_{i,t}(K) - \boldsymbol{\mu}_t(K)}_{\text{variance}}$$

$$\boldsymbol{\mu}_t(K) = \mathbb{E}_i[\boldsymbol{h}_{i,t}(K)]$$

$$\boldsymbol{\mu}_t(\infty) = \lim_{K\to\infty} \mathbb{E}_i[\boldsymbol{h}_{i,t}(K)]$$

# Decrease of Bias and Variance




Decrease of Bias:

$$\frac{\|\boldsymbol{\mu}_t(K) - \boldsymbol{\mu}_t(\infty)\|_2}{\|\boldsymbol{\mu}_t(0) - \boldsymbol{\mu}_t(\infty)\|_2} \propto \mathcal{O}\left(\frac{1}{K}\right)$$

Decrease of Variance:

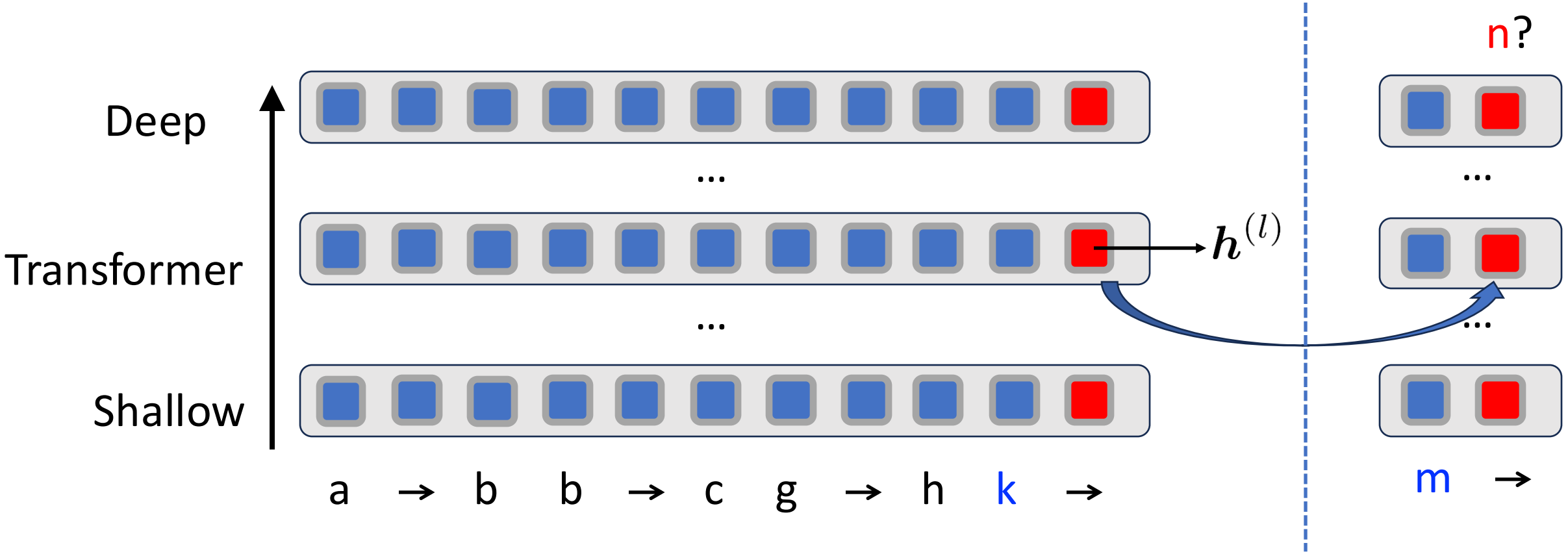
$$\left\|\mathbb{E}\left[\left(\mathbf{h}_{t,i}(K) - \boldsymbol{\mu}_t(K)\right)^2\right]\right\|_2 \propto \mathcal{O}\left(\tfrac{1}{K}\right)$$

- A formal analysis for simplified models (linear attention)

Jiachen Jiang, Yuxin Dong, Jinxin Zhou, Zhihui Zhu, From Compression to Expression: A Layerwise Analysis of In-Context Learning, 2025

# Task Vector Accuracy

- Query + task-vector $\boldsymbol{h}^{(l)}$ is enough to perform ICL

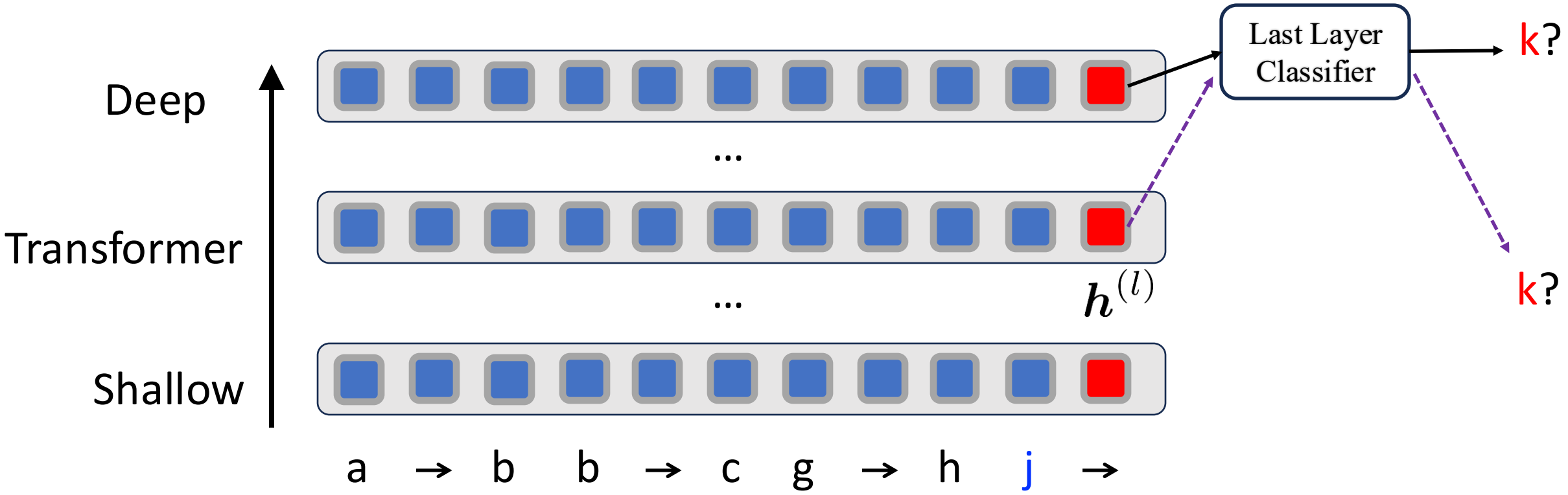



- How much the hidden representation captures the task information

Hendel, Geva, Globerson, In-Context Learning Creates Task Vectors, 2023.
S Liu, H Ye, L Xing, J Zou, In-context vectors: Making in context learning more effective and controllable through latent space steering, ICML 2024.

# Early Exit Accuracy

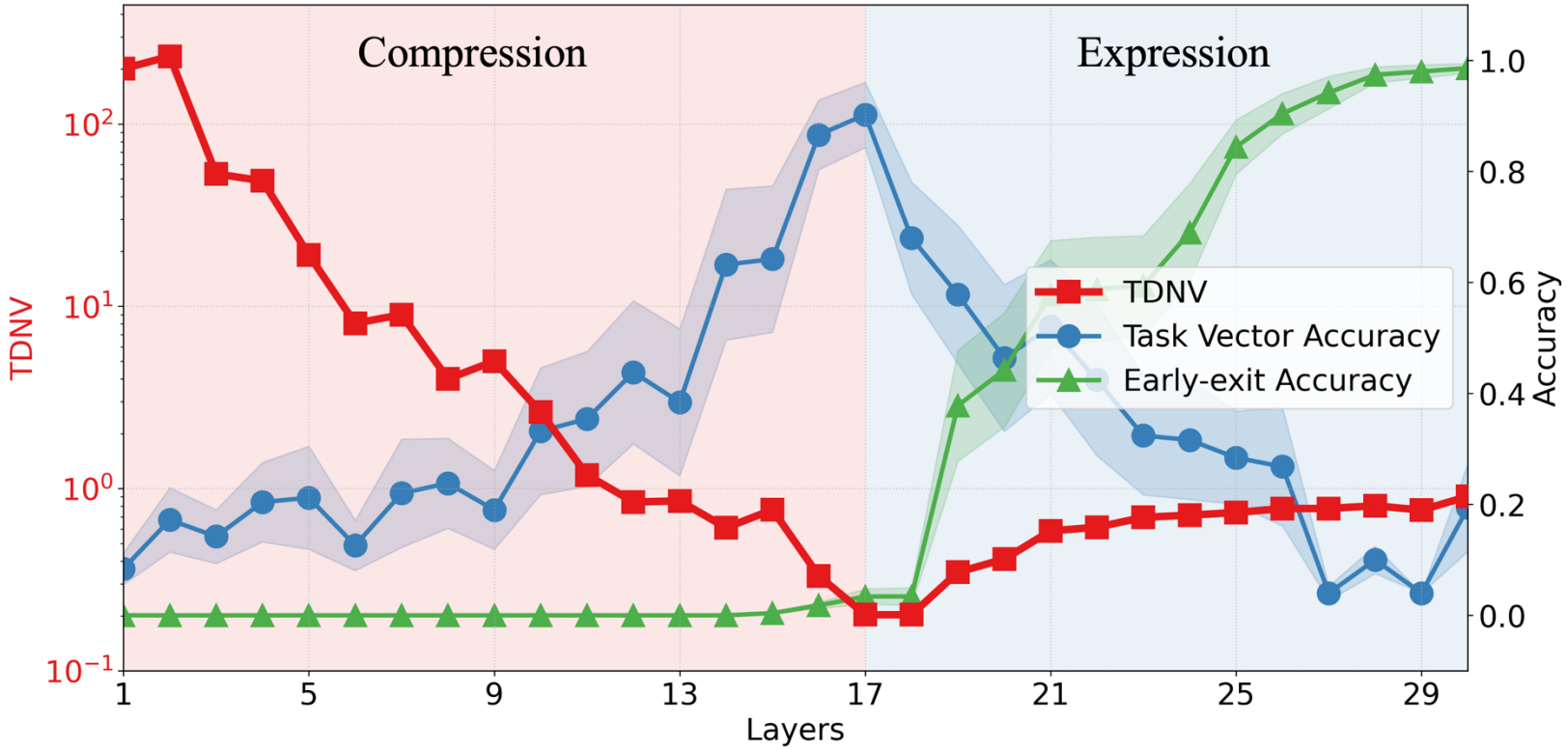
Make prediction based on intermediate layer hidden states instead of last layer




- How much the hidden representation captures the query information

DeeBERT: Dynamic Early Exiting for Accelerating BERT Inference. ACL 2020.

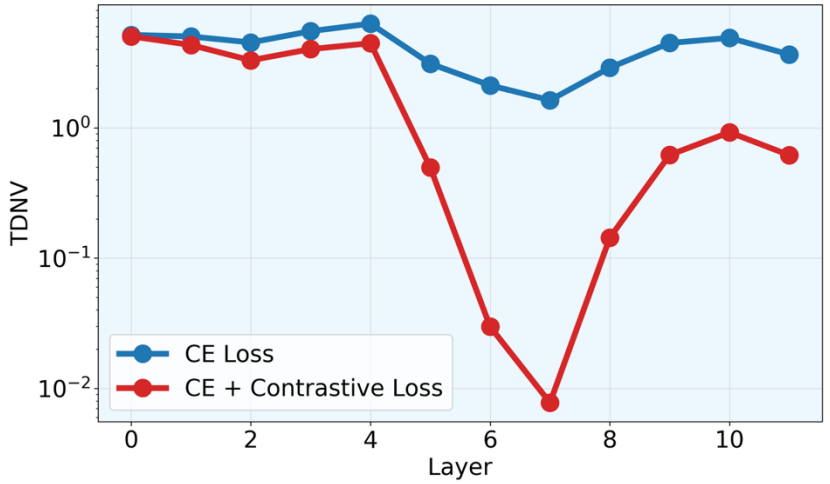
# Layerwise Compression and Expression

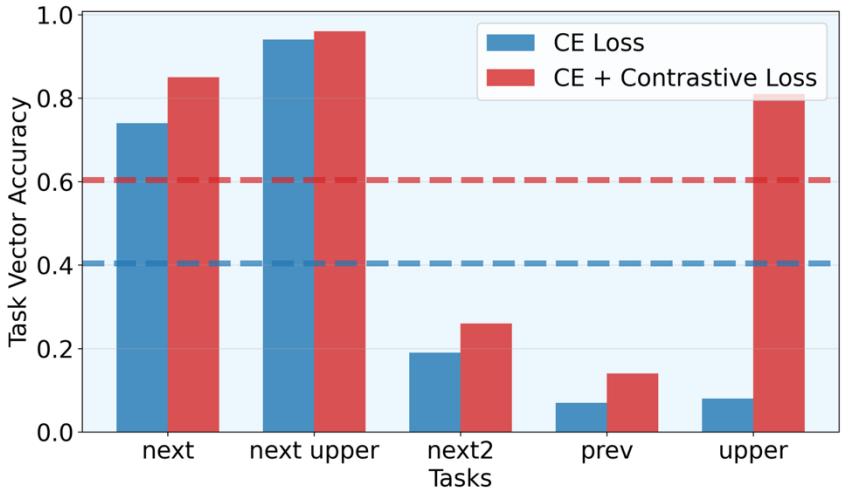



- The most compact representation (***smallest*** TDNV) achieves ***best*** **Task Vector Accuracy** (Query + task-vector $\boldsymbol{h}^{(l)}$)
- **Early-exit Accuracy** starts increasing after the most compact layer

# Promoting Compression Improves Performance

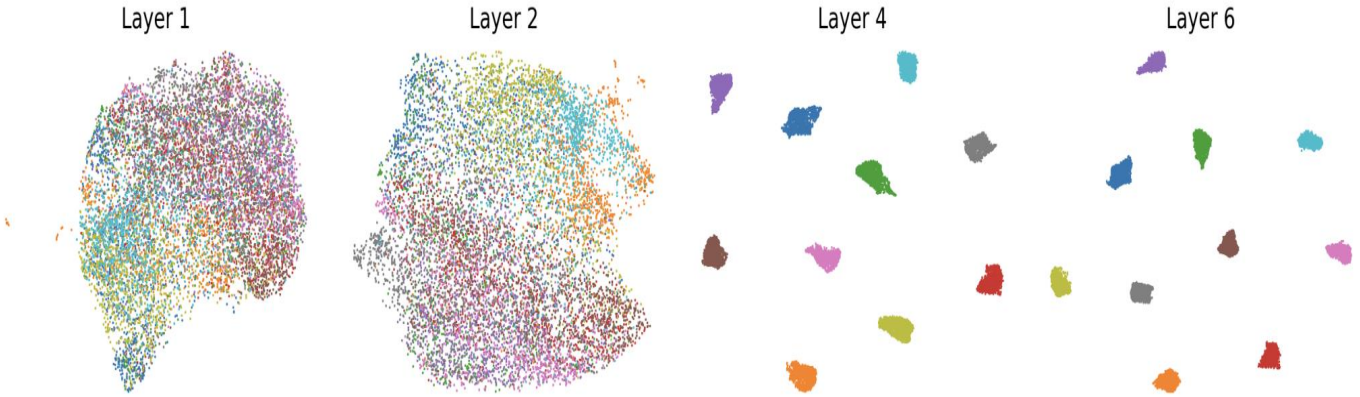
Task-vector contrastive fine-tuning improves task-vector accuracy

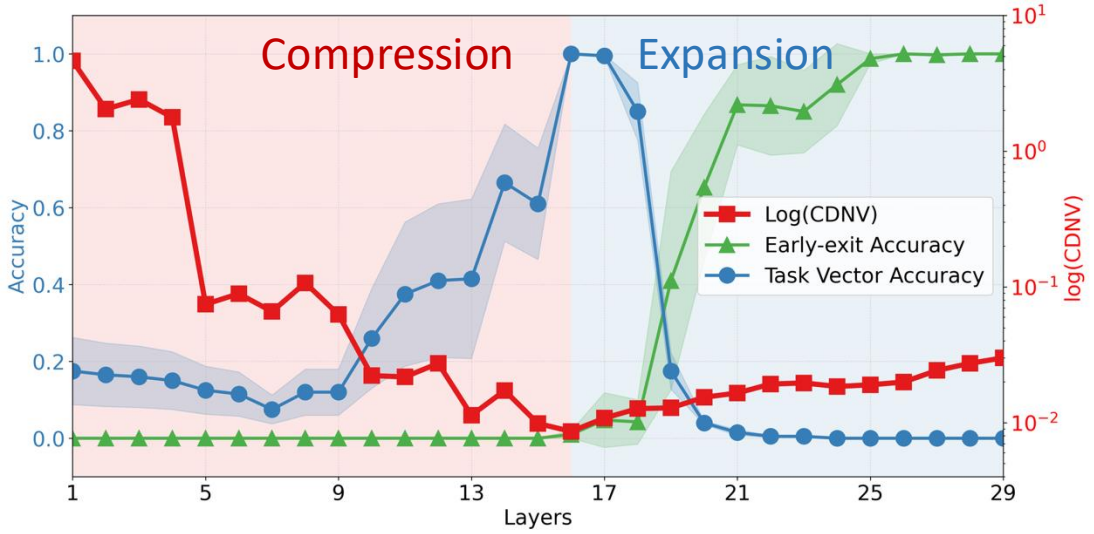



- **Method**: during fine-tuning on ICL tasks, add contrastive loss on intermediate layer features to encourage compression
- **Results**: more compressed representations yield better performance; task-vector accuracy improves by 20%

Linear, compact, and structured representation emerges in deep classifiers and (multimodal) large language models




**The objective of learning**: Transform nonlinear and complex data to a linear, compact, and structured representation.

**The effect of depth**: Progressively achieve this goal through layer-wise transformation

# This Tutorial: The Outline

- Session I: Introduction of Basic Low-dimensional Models
- Session II: Understanding Low-Dimensional Structures in Representation Learning
  - Lec 2.1: Bridging Symbolic Abstraction and Low-Dimensionality in Machine Reasoning: Algebraic and Geometric Perspectives
  - **Lec 2.2: Emergency of Low-dimensional Representations in Deep Models**
- Session III: Understanding Low-Dimensional Structures in **Diffusion Generative Models** (**starts from 1PM)**
  - Lec 3.1: Low-Dimensional Models for Understanding Generalization in Diffusion Models
  - Lec 3.2: Explore Low-Dimensional Structures for Constrained and Controllable Diffusion Models in Scientific Applications
- Session IV: **Designing** Deep Networks for Pursuing Low-Dimensional Structures
  - Lec 4.1: ReduNet: A White-box Deep Network from the Principle of Maximizing Rate Reduction
  - Lec 4.2: White-Box Transformers via Sparse Rate Reduction
- Session V: Panel Discussion: Sara Fridovich-Keil, Berivan Isik, Vladimir Pavlovic