DEMYSTIFYING OPTIMIZATION AND GENERALIZATION OF DEEP LEARNING VIA FEATURE LEARNING THEORY

Andi Han (Lecturer, University of Sydney), Wei Huang (Research Scientist, RIKEN AIP & ISM)

Tutorial @ AJCAI 2025, Canberra

SIGNIFICANT SUCCESS OF DEEP LEARNING

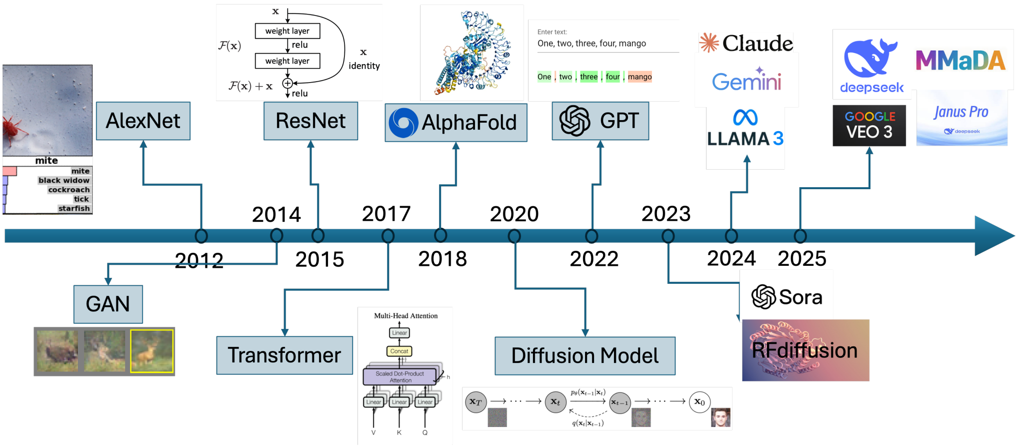


Image credits (left → right): Krizhevsky et al. 2012; Goodfellow et al. 2014; He et al. 2015; Vaswani et al. 2017; https://alphafold.ebi.ac.uk/; Ho et al. 2020;https://huggingface.co/blog/alonsosilva/nexttokenprediction; Watson et al. 2023

MODEL SIZE AND COMPLEXITY GROWTH

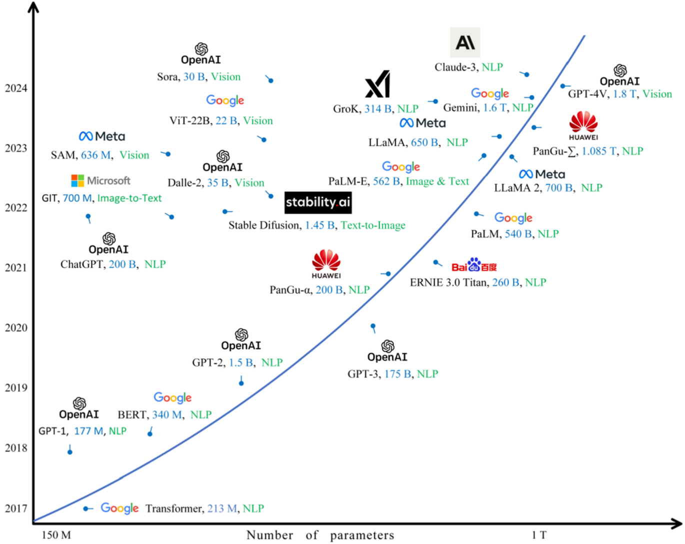


Image credits: Tu et al. 2024

YET WE UNDERSTAND LESS AND LESS...

- Transparency
- Robustness
- Privacy, Fairness, Biases

xkcd: machine learning

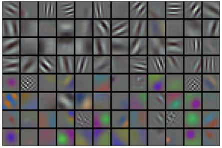
YET WE UNDERSTAND LESS AND LESS...

- Transparency
- Robustness
- Privacy, Fairness, Biases

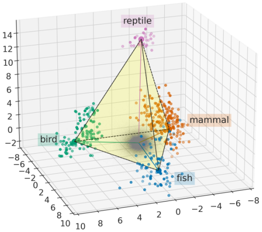
Underlying principles of deep learning?

xkcd: machine learning

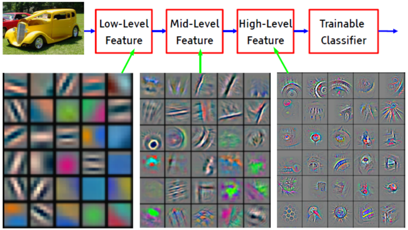
FEATURE LEARNING AT THE CORE OF DEEP LEARNING



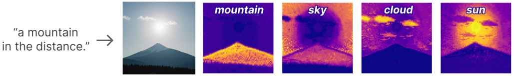
AlexNet 1st layer (Krizhevsky et al. 2012)



Concept features in Gemma (Park et al. 2025)



Hierarchical feature learning (LeCun 2015)



Cross-modal features (Helbling et al. 2025)

TUTORIAL BREAKDOWN

Goal

- Introduce a theoretical sandbox to understand deep learning via feature learning
- Bridge empirical phenomena and theoretical insights on optimization and generalization

Outline

1. Deep learning benefits from feature learning
2. A signal-and-noise data model
3. Benign Overfitting with Feature Learning
4. Feature Learning under Different Training Strategies
4. Feature Learning in Foundation Generative Models
5. Conclusions & outlook

WHY DEEP LEARNING BENEFITS FROM FEATURE LEARNING

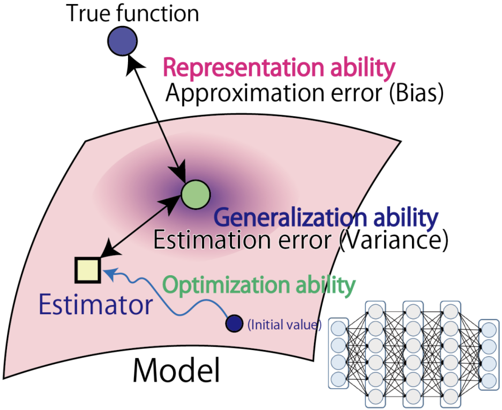
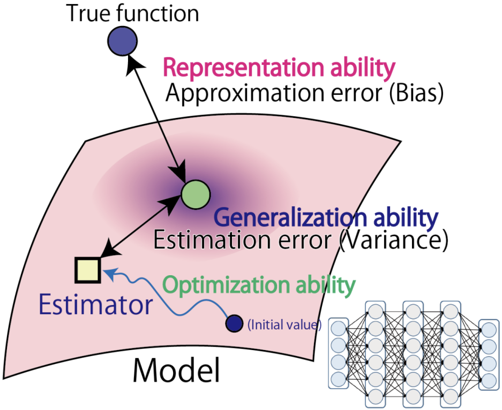


Image credits: Suzuki 2024

WHY DEEP LEARNING BENEFITS FROM FEATURE LEARNING



Feature Learning Affects ALL!

Image credits: Suzuki 2024

A REPRESENTATION PERSPECTIVE

Two-layer Neural Network and Kernel

$$f(\boldsymbol{x}) = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} a_i \sigma(\langle \boldsymbol{w}_i, \boldsymbol{x} \rangle)$$

A REPRESENTATION PERSPECTIVE

Two-layer Neural Network and Kernel

$$f(\boldsymbol{x}) = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} a_i \sigma(\langle \boldsymbol{w}_i, \boldsymbol{x} \rangle)$$

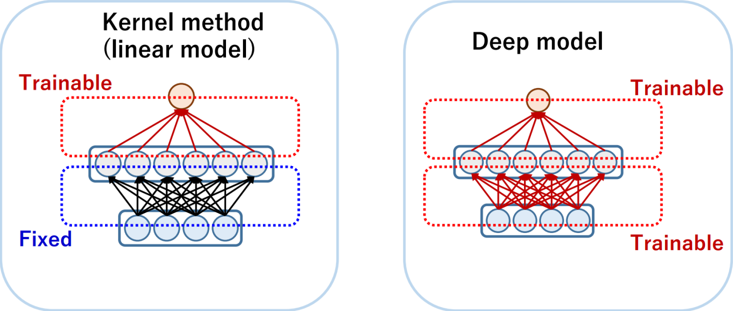


Image credits: Suzuki 2024

A REPRESENTATION PERSPECTIVE

Target function $f^*(\boldsymbol{x}) = \sigma^*(\langle \boldsymbol{x}, \boldsymbol{\beta}_* \rangle)$. $y_i = f^*(\boldsymbol{x}_i) + \epsilon_i$, $\boldsymbol{x}_i \sim \mathcal{N}(0, \mathbf{I})$, $\epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2)$.
Let $\hat{f}_\lambda = \arg\min_f \frac{1}{n} \sum_{i=1}^n (f(\boldsymbol{x}_i) - y_i)^2 + \frac{\lambda}{N} \|\boldsymbol{a}\|^2$.

- (*Kernel*) Let $\boldsymbol{w}_i$ fixed at initialization $\boldsymbol{w}_i^0$

$$\inf_\lambda \mathcal{R}(\hat{f}_\lambda) \geq \|\mathsf{P}_{>1} f^*\|_{L^2}^2 + o(1)$$

- (*Neural Network*) Let $\boldsymbol{w}_i$ be one-step gradient update from $\boldsymbol{w}_i^0$,*

$$\mathcal{R}(\hat{f}_\lambda) < \|\mathsf{P}_{>1} f^*\|_{L^2}^2$$

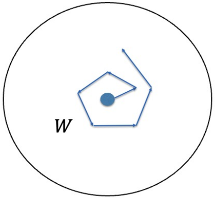
where $\mathcal{R}(\hat{f}) = \mathbb{E}_{\boldsymbol{x}}(\hat{f}(\boldsymbol{x}) - f^*(\boldsymbol{x}))^2$ is the prediction risk, and $f^*(\boldsymbol{x}) = \mu_0^* + \mu_1^* \langle \boldsymbol{x}, \boldsymbol{\beta}_* \rangle + \mathsf{P}_{>1} f^*$.

Ba et al. 2022. "High-dimensional Asymptotics of Feature Learning: How One Gradient Step Improves the Representation".

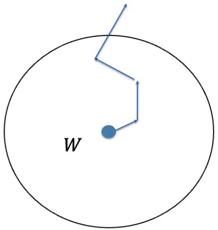
* *Suppose* $\sigma = \sigma^* = \tanh$.

A REPRESENTATION PERSPECTIVE

Feature Learning underlies the success of deep learning!



Lazy training



Feature learning

$$\|W(t) - W(0)\|_F = O\left(\frac{1}{\sqrt{N}}\right) \qquad \|W(t) - W(0)\|_F = \Omega(1)$$

SIGNAL-NOISE DATA MODEL (A SANDBOX FOR FEATURE LEARNING)

Feature Decomposition: Data $\approx$ Signal + Noise

SIGNAL-NOISE DATA MODEL (A SANDBOX FOR FEATURE LEARNING)

Feature Decomposition: Data $\approx$ Signal + Noise

Signal-noise data model (Cao et al., 2022; Kou et al., 2023)

Data $\boldsymbol{x} = [y\boldsymbol{\mu}, \boldsymbol{\xi}]$

- (Signal) $\boldsymbol{\mu}$ is a signal vector, and y is the label
- (Noise) $\boldsymbol{\xi}$ is a random noise (commonly assumed to be Gaussian $\mathcal{N}(\mathbf{0}, \sigma_\xi^2 \mathbf{I})$)

SIGNAL-NOISE DATA MODEL (A SANDBOX FOR FEATURE LEARNING)

Feature Decomposition: Data $\approx$ Signal + Noise

Signal-noise data model (Cao et al., 2022; Kou et al., 2023)

Data $\boldsymbol{x} = [y\boldsymbol{\mu}, \boldsymbol{\xi}]$

- (Signal) $\boldsymbol{\mu}$ is a signal vector, and y is the label
- (Noise) $\boldsymbol{\xi}$ is a random noise (commonly assumed to be Gaussian $\mathcal{N}(\mathbf{0}, \sigma_{\xi}^2\mathbf{I})$)

Demo from Imagenet

Today is Sunday

Signal

Noise

BENIGN OVERFITTING WITH FEATURE LEARNING

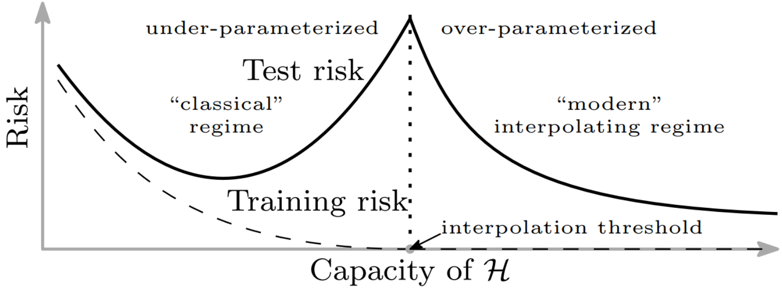


Image credit: Belkin et al. 2019. "Reconciling modern machine learning practice and the bias-variance trade-off"

Benign Overfitting with Feature Learning

BENIGN OVERFITTING WITH FEATURE LEARNING

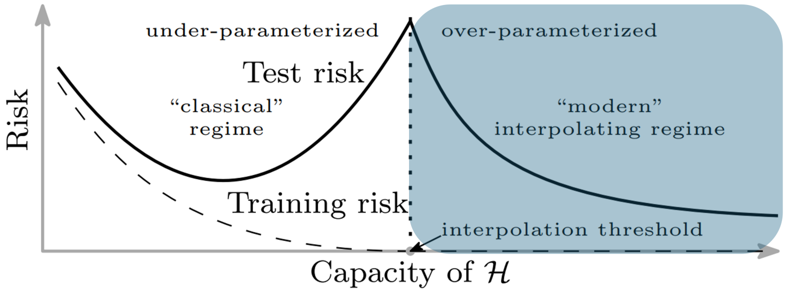


Image credit: Belkin et al. 2019. "Reconciling modern machine learning practice and the bias-variance trade-off"

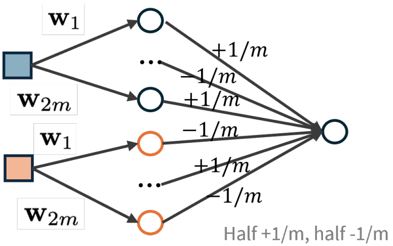
BENIGN OVERFITTING WITH FEATURE LEARNING (MODEL SETUP)

Two-layer Convolutional Neural Network

$$f(\mathbf{W}, \boldsymbol{x}) = F_1(\mathbf{W}_1, \boldsymbol{x}) - F_{-1}(\mathbf{W}_{-1}, \boldsymbol{x})$$

$$\text{where } F_j(\mathbf{W}_j, \boldsymbol{x}) = \frac{1}{m}\sum_{r=1}^{m} \sigma\big(\langle \boldsymbol{w}_{j,r}, \boldsymbol{x}^{(1)}\rangle\big) + \frac{1}{m}\sum_{r=1}^{m} \sigma\big(\langle \boldsymbol{w}_{j,r}, \boldsymbol{x}^{(2)}\rangle\big)$$

where $\boldsymbol{x} = [\boldsymbol{x}^{(1)\top}, \boldsymbol{x}^{(2)\top}]^\top = [y\boldsymbol{\mu}, \boldsymbol{\xi}]$



BENIGN OVERFITTING WITH FEATURE LEARNING (TRAINING SETUP)

- Training Data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$:
  - Binary classification: $y = \pm 1$ with equal chance
  - $\boldsymbol{x}_i = [y_i \boldsymbol{\mu}, \boldsymbol{\xi}_i]$, with fixed signal $\boldsymbol{\mu}$ and random noise $\boldsymbol{\xi}_i \sim \mathcal{N}\big(\mathbf{0}, \sigma_\xi^2 \mathbf{I}_d\big)$
  - Define SNR$= \|\boldsymbol{\mu}\|/(\sigma_\xi \sqrt{d})$
- Training Loss:

$$L_S(\mathbf{W}) = \frac{1}{n}\sum_{i=1}^n \ell\big(y_i f(\mathbf{W}, \boldsymbol{x}_i)\big), \qquad \ell(z) = \log\big(1 + \exp(-z)\big)$$

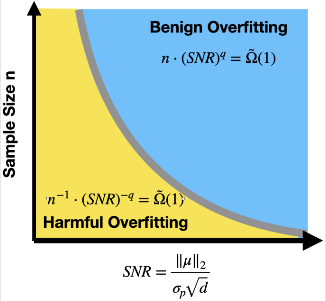
- Test Loss: $L_D(\mathbf{W}) = \mathbb{E}_{(\boldsymbol{x},y)}\big[\ell\big(y f(\mathbf{W}, \boldsymbol{x})\big)\big]$.
- Training Algorithm: $\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} - \eta \nabla L_S(\mathbf{W}), \qquad \mathbf{W}^{(0)} \sim \mathcal{N}(\mathbf{0}, \sigma_0^2 \mathbf{I})$

BENIGN OVERFITTING WITH FEATURE LEARNING (MAIN RESULTS)

Feature learning under (ReLU)^q (Cao et al., 2022, Theorem 4.3 & 4.4)

Suppose $\sigma = (\text{ReLU})^q$, $(q > 2)$, under *over-parameterization*[a] and *small initialization*[b], there exists an iterate $\mathbf{W}^{(t)}$ with $L_S(\mathbf{W}^{(t)}) \leq \varepsilon$ and

- Benign Overfitting: When $n \cdot \text{SNR}^q = \widetilde{\Omega}(1)$,
  - $L_D(\mathbf{W}^{(t)}) \leq 6\varepsilon + \exp(-n^2)$
- Harmful Overfitting: When $n^{-1} \cdot \text{SNR}^{-q} = \widetilde{\Omega}(1)$,
  - $L_D(\mathbf{W}^{(t)}) \geq 0.1$



[a] Large d relative to n [b] Feature learning regime

BENIGN OVERFITTING WITH FEATURE LEARNING (KEY IDEA)

Signal-Noise Decomposition for feature learning

$$\boldsymbol{w}_{j,r}^{(t)} = \boldsymbol{w}_{j,r}^{(0)} + j \cdot \gamma_{j,r}^{(t)} \cdot \|\boldsymbol{\mu}\|^{-1} \cdot \boldsymbol{\mu} + \sum_{i=1}^{n} \rho_{j,r,i}^{(t)} \cdot \|\boldsymbol{\xi}_i\|^{-2} \cdot \boldsymbol{\xi}_i$$

such that $\gamma_{j,r}^{(t)} \approx \langle \boldsymbol{w}_{j,r}^{(t)}, \boldsymbol{\mu} \rangle$ (signal learning) and $\rho_{j,r,i}^{(t)} \approx \langle \boldsymbol{w}_{j,r}^{(t)}, \boldsymbol{\xi}_i \rangle$ (noise memorization)

- Benign Overfitting: learn signal and ignore noise

$$\max_r \gamma_{j,r}^{(t)} \geq C_1 > 0, \quad \max_{j,r,i} |\rho_{j,r,i}^{t}| \approx 0 \qquad \text{(Signal dominates ☺)}$$

- Harmful Overfitting: memorize noise and ignore signal

$$\max_r \rho_{y_i,r,i}^{(t)} \geq C_2 > 0, \quad \max_{j,r} \gamma_{j,r}^{(t)} \approx 0 \qquad \text{(Noise dominates ☹)}$$

BENIGN OVERFITTING WITH FEATURE LEARNING (RELU)

Feature learning with ReLU (Kou et al., 2023, Theorem 4.2)

Suppose $\sigma = \mathrm{ReLU}$, under *over-parameterization*[a] and *small initialization*[b], and $n \cdot \mathrm{SNR}^2 = o(1)$, there exists an iterate $\mathbf{W}^{(t)}$ with $L_S(\mathbf{W}^{(t)}) \leq \varepsilon$:

- Benign Overfitting: When $n\|\boldsymbol{\mu}\|^4 \geq C_1 \sigma_\xi^4 d$,
  - $L_D^{0-1}(\mathbf{W}^{(t)}) \leq \exp\big(-n\|\boldsymbol{\mu}\|^4/(C_2\sigma_\xi^4 d)\big)$
- Harmful Overfitting: When $n\|\boldsymbol{\mu}\|^4 \leq C_3 \sigma_\xi^4 d$,
  - $L_D^{0-1}(\mathbf{W}^{(t)}) \geq 0.1$

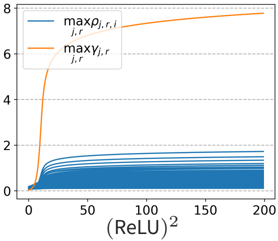
where $L_D^{0-1}(\mathbf{W}) = \mathbb{P}_{(\boldsymbol{x},y)}[y \neq \mathrm{sign}\big(f(\mathbf{W}, \boldsymbol{x})\big)]$ is the test error.

Key differences to $(\mathrm{ReLU})^q$: Constant separation and low-SNR regime.

BENIGN OVERFITTING WITH FEATURE LEARNING (A COMPARISON OF FEATURE LEARNING)

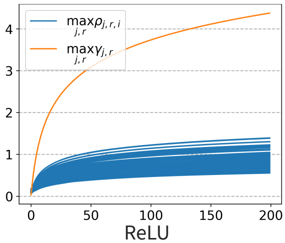
$(\mathrm{ReLU})^q$: Polynomial growth

$$\gamma_{j,r}^{(t+1)} \approx (1-\eta_\gamma)(\gamma_{j,r}^{(t)})^{q-1}$$
$$\rho_{j,r,i}^{(t+1)} \approx (1-\eta_\xi)(\rho_{j,r,i}^{(t)})^{q-1}$$



ReLU: Linear growth

$$\gamma_{j,r}^{(t+1)} \approx \gamma_{j,r}^{(t+1)} + \eta'_\gamma$$
$$\rho_{j,r,i}^{(t+1)} \approx \rho_{j,r,i}^{(t)} + \eta'_\xi$$



FEATURE LEARNING WITH LABEL NOISE

Label noise is common: the observed label $\tilde{y}$ is not equal to the ground-truth label y!

Dataset	**MNIST [26]**	**CIFAR-10 [27]**	**CIFAR-100 [36]**
Instance			
Given Label	"8"	"cat"	"Lobstar"
Actual Label	"9"	"frog"	"Crab"
Dataset	**Caltech-256 [37]**	**ImageNet [38]**	**QuickDraw [39]**
Instance			
Given Label	"Dolphin"	"White stork"	"Tiger"
Actual Label	"Kayak"	"Black stork"	"eye"

Image credit: Bhatt et al. 2024

BENIGN OVERFITTING WITH FEATURE LEARNING (LABEL NOISE)

Feature learning under label noise (Han et al., 2025b, Theorem 4.2 & 4.4)

Observed label $\tilde{y} \neq y$ (with $\mathbb{P}(\tilde{y} \neq y) = \tau$), and $\sigma = \text{ReLU}$, $n \cdot \mathrm{SNR}^2 = \Theta(1)$

Two-stage behavior

BENIGN OVERFITTING WITH FEATURE LEARNING (LABEL NOISE)

Feature learning under label noise (Han et al., 2025b, Theorem 4.2 & 4.4)

Observed label $\tilde{y} \neq y$ (with $\mathbb{P}(\tilde{y} \neq y) = \tau$), and $\sigma = \mathrm{ReLU}$, $n \cdot \mathrm{SNR}^2 = \Theta(1)$

Two-stage behavior

- Stage I (Model fits clean data): there exists T_1 s.t.
  - Model learns more signal than noise, i.e., $\gamma_{j,r}^{(T_1)} > \rho_{\tilde{y}_i,r,i}^{(T_1)}$
  - For all clean samples: $\tilde{y}_i f(\mathbf{W}^{(T_1)}, \boldsymbol{x}_i) \geq 0$. For all noisy samples: $\tilde{y}_i f(\mathbf{W}^{(T_1)}, \boldsymbol{x}_i) < 0$
  - Early stopping works: $L_D^{0-1}(\mathbf{W}^{(T_1)}) \leq \exp(-\Omega(d/n))$

BENIGN OVERFITTING WITH FEATURE LEARNING (LABEL NOISE)

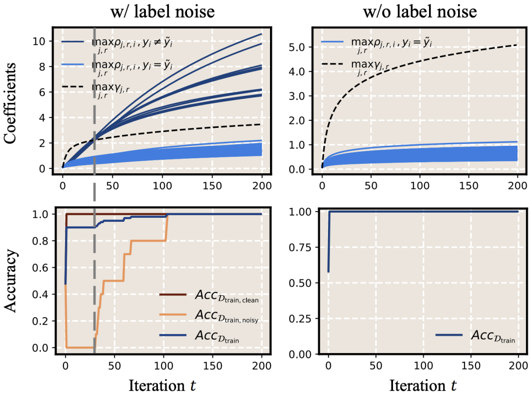
Feature learning under label noise (Han et al., 2025b, Theorem 4.2 & 4.4)

Observed label $\tilde{y} \neq y$ (with $\mathbb{P}(\tilde{y} \neq y) = \tau$), and $\sigma = \text{ReLU}$, $n \cdot \mathrm{SNR}^2 = \Theta(1)$

Two-stage behavior

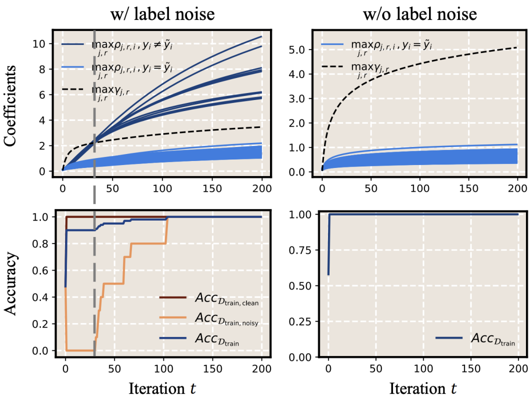
- Stage I (Model fits clean data): there exists T_1 s.t.
  - Model learns more signal than noise, i.e., $\gamma_{j,r}^{(T_1)} > \rho_{\tilde{y}_i,r,i}^{(T_1)}$
  - For all clean samples: $\tilde{y}_i f(\mathbf{W}^{(T_1)}, \boldsymbol{x}_i) \geq 0$. For all noisy samples: $\tilde{y}_i f(\mathbf{W}^{(T_1)}, \boldsymbol{x}_i) < 0$
  - Early stopping works: $L_D^{0-1}(\mathbf{W}^{(T_1)}) \leq \exp(-\Omega(d/n))$
- Stage II (Model overfits noisy data): there exists $t^* \geq T_1$ such that
  - For most if not all samples: $\tilde{y}_i f(\mathbf{W}^{(T_1)}, \boldsymbol{x}_i) \geq 0$.
  - Model fails to generalize: $L_D^{0-1}(\mathbf{W}^{(t^*)}) \geq 0.5\tau$

BENIGN OVERFITTING WITH FEATURE LEARNING (LABEL NOISE)

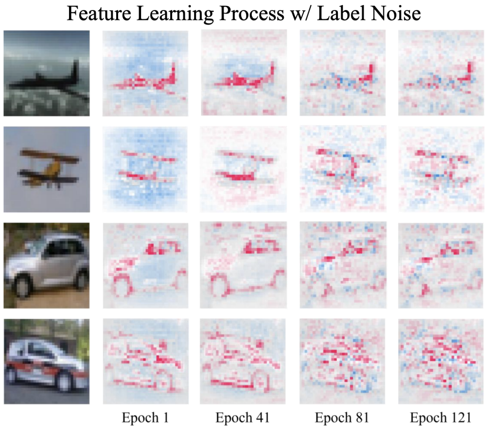


Synthetic data

BENIGN OVERFITTING WITH FEATURE LEARNING (LABEL NOISE)



Synthetic data



VGG on CIFAR-10 with label noise.

BENIGN OVERFITTING WITH FEATURE LEARNING (TRANSFORMER)

The emergence of large language models (LLMs) is due to Transformers.

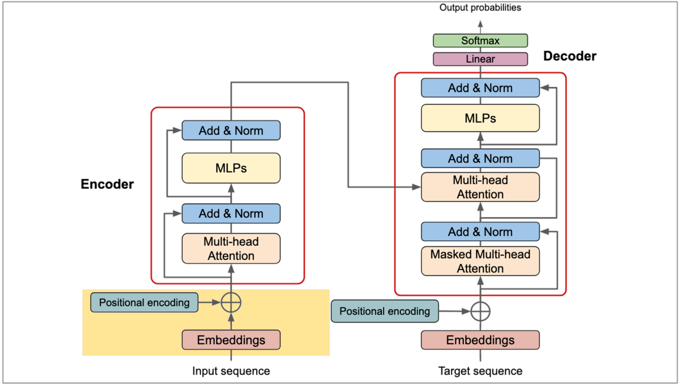


Image credit: https://deeprevision.github.io/posts/001-transformer/

BENIGN OVERFITTING WITH FEATURE LEARNING (TRANSFORMER)

To analyze benign overfitting, focus on the core mechanism: attention.

- Consider a single-head Transformer with *global average pooling* (Jiang et al., 2024):

$$f(\mathbf{X}) = \underbrace{\frac{1}{L}\sum_{\ell=1}^{L}}_{\text{Avg Pooling}} \underbrace{\mathrm{Softmax}(\boldsymbol{x}^{(\ell)}\mathbf{W}_Q\mathbf{W}_K^\top\mathbf{X}^\top)}_{\text{Attention for token } l}\mathbf{X}\mathbf{W}_V\boldsymbol{w}_o$$

  where $\mathbf{X} = [\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, ..., \boldsymbol{x}^{(L)}]^\top \in \mathbb{R}^{L\times d}$.

Training data $(\mathbf{X}_i, y_i)$:

- $\boldsymbol{x}^{(1)} = y\boldsymbol{\mu}$ (signal), $\boldsymbol{x}^{(2)}, ..., \boldsymbol{x}^{(L)} \sim \mathcal{N}(\mathbf{0}, \sigma_\xi^2\mathbf{I})$ (noise).

Feature learning in Transformers (Jiang et al., 2024, Theorem 4.1 & 4.2)

There exists T, s.t. $L_S(\mathbf{\Theta}^{(T)}) \approx 0$ and

Feature learning in Transformers (Jiang et al., 2024, Theorem 4.1 & 4.2)

There exists T, s.t. $L_S(\mathbf{\Theta}^{(T)}) \approx 0$ and

- Benign overfitting: under condition $n \cdot \mathrm{SNR}^2 = \Omega(1)$
  - Attention on *signal*: $\langle \mathbf{W}_Q^{(T)} \boldsymbol{x}^{(\ell)}, \mathbf{W}_K^{(T)} \boldsymbol{\mu} \rangle = \Omega(1)$, $\langle \mathbf{W}_Q^{(T)} \boldsymbol{x}^{(\ell)}, \mathbf{W}_K^{(T)} \boldsymbol{\xi} \rangle \approx 0$
  - Value focuses on *signal*: $\langle \mathbf{W}_V^{(T)} \boldsymbol{w}_o, \boldsymbol{\mu} \rangle > \langle \mathbf{W}_V^{(T)} \boldsymbol{w}_o, \boldsymbol{\xi} \rangle$
  - The test loss is nearly zero: $L_D(\mathbf{\Theta}^{(T)}) \approx 0$

BENIGN OVERFITTING WITH FEATURE LEARNING (TRANSFORMER)

Feature learning in Transformers (Jiang et al., 2024, Theorem 4.1 & 4.2)

There exists T, s.t. $L_S(\boldsymbol{\Theta}^{(T)}) \approx 0$ and

- Benign overfitting: under condition $n \cdot \mathrm{SNR}^2 = \Omega(1)$
  - Attention on *signal*: $\langle \mathbf{W}_Q^{(T)} \boldsymbol{x}^{(\ell)}, \mathbf{W}_K^{(T)} \boldsymbol{\mu} \rangle = \Omega(1)$, $\langle \mathbf{W}_Q^{(T)} \boldsymbol{x}^{(\ell)}, \mathbf{W}_K^{(T)} \boldsymbol{\xi} \rangle \approx 0$
  - Value focuses on *signal*: $\langle \mathbf{W}_V^{(T)} \boldsymbol{w}_o, \boldsymbol{\mu} \rangle > \langle \mathbf{W}_V^{(T)} \boldsymbol{w}_o, \boldsymbol{\xi} \rangle$
  - The test loss is nearly zero: $L_D(\boldsymbol{\Theta}^{(T)}) \approx 0$
- Harmful overfitting: under condition $n^{-1} \cdot \mathrm{SNR}^{-2} = \Omega(1)$
  - Attention on *noise*: $\langle \mathbf{W}_Q^{(T)} \boldsymbol{x}^{(\ell)}, \mathbf{W}_K^{(T)} \boldsymbol{\xi} \rangle = \Omega(1)$, $\langle \mathbf{W}_Q^{(T)} \boldsymbol{x}^{(\ell)}, \mathbf{W}_K^{(T)} \boldsymbol{\mu} \rangle \approx 0$
  - Value focuses on *noise*: $\langle \mathbf{W}_V^{(T)} \boldsymbol{w}_o, \boldsymbol{\xi} \rangle > \langle \mathbf{W}_V^{(T)} \boldsymbol{w}_o, \boldsymbol{\mu} \rangle$
  - The test loss is high: $L_D(\boldsymbol{\Theta}^{(T)}) = \Theta(1)$

OTHER RELEVANT WORKS

Meng et al. 2024. Benign overfitting in two-layer ReLU convolutional neural networks for XOR data. *International Conference on Machine Learning* (*ICML 2025*).

Huang et al. 2025. Quantifying the Optimization and Generalization Advantages of Graph Neural Networks Over Multilayer Perceptrons. *International Conference on Artificial Intelligence and Statistics (AISTATS 2025)*.

Karhadkar et al. 2024. Benign overfitting in leaky relu networks with moderate input dimension. *Advances in Neural Information Processing Systems (NeurIPS 2024).*

Shang et al. 2024. Initialization Matters: On the Benign Overfitting of Two-Layer ReLU CNN with Fully Trainable Layers. *arXiv preprint arXiv:2410.19139*.

Sakamoto & Sato. 2025. Benign Overfitting in Token Selection of Attention Mechanism. *International Conference on Machine Learning (ICML 2025)*.

Frei et al. 2022. Benign overfitting without linearity: Neural network classifiers trained by gradient descent for noisy linear data. *Conference on Learning Theory (COLT 2022).*

Feature Learning under Different Training Strategies

FEATURE LEARNING UNDER DIFFERENT TRAINING STRATEGY

Optimization and Generalization are entangled in Deep Learning

FEATURE LEARNING UNDER DIFFERENT TRAINING STRATEGY

Optimization and Generalization are entangled in Deep Learning

- A tweak in training strategy can *drastically* affect convergence in training and test error

FEATURE LEARNING UNDER DIFFERENT TRAINING STRATEGY

Optimization and Generalization are entangled in Deep Learning

- A tweak in training strategy can *drastically* affect convergence in training and test error

Adaptive Gradient

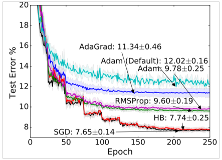


Image credit: Wilson et al. (2017)

Sharpness Aware Minimization

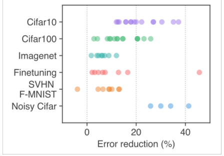


Image credit: Foret et al. (2021)

Momentum

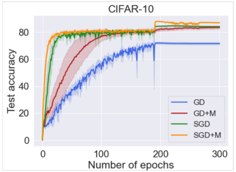


Image credit: Jelassi & Li (2022)

Mixup

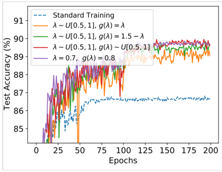


Image credit: Zou et al. (2022)

Large Learning Rate

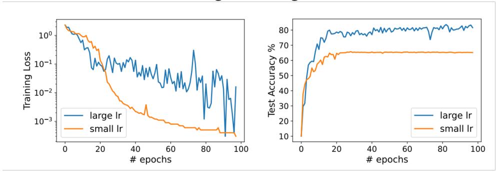


Image credit: Lu et al. (2023)

Label Noise Regularization

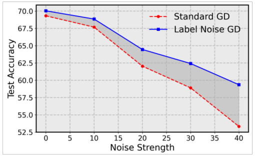


Image credit: Huang et al. (2025)

Matrix Normalization

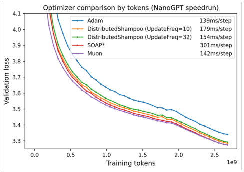


Image credit: Jordan (2025)

FEATURE LEARNING WITH SIGN GD AND ADAM

Adam (Kigma & Ba 2015)

$$\boldsymbol{m}_t = \beta_1 \boldsymbol{m}_{t-1} + (1-\beta_1)\boldsymbol{g}_t$$

$$\boldsymbol{v}_t = \beta_2 \boldsymbol{v}_{t-1} + (1-\beta_2)\boldsymbol{g}_t \odot \boldsymbol{g}_t$$

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \frac{\boldsymbol{m}_t}{\sqrt{\boldsymbol{v}_t} + \epsilon}$$

Sign-GD (Adam when $\beta_1 = \beta_2 = 0$)

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \mathrm{sign}(\boldsymbol{g}_t)$$

Sign GD *close* to Adam

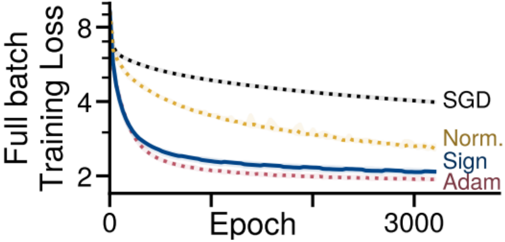


Image Credit: Kunstner et al. 2023.

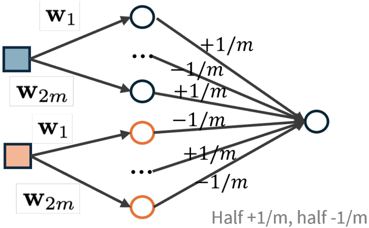
A *sparse* signal-noise model (Zou et al., 2023)

Data $\boldsymbol{x} = [y\boldsymbol{\mu}, \boldsymbol{\xi}]$

- (Signal) $\boldsymbol{\mu} = [1, 0, 0, ..., 0]^\top$
- (Noise) $\boldsymbol{\xi} = \widetilde{\boldsymbol{\xi}} \odot \boldsymbol{s}$, where $\widetilde{\boldsymbol{\xi}} \sim \mathcal{N}(\mathbf{0}, \sigma_\xi^2 \mathbf{I}_d)$, $\boldsymbol{s} \in \{0, 1\}^d$ is a random binary mask[a]

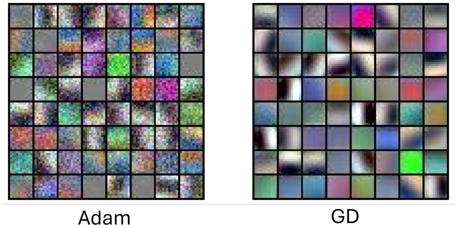
[a] Further adversarial feature noise is added

Consider the same two-layer CNN (with $\sigma = \mathrm{ReLU}^q$ $(q \geq 3)$)



Adam generalizes worse than GD (Zou et al., 2023, Theorem 4.1)

- Adam can output a stationary point $\mathbf{W}_{\text{adam}}$ in L_1 norm with
  - $L_S(\mathbf{W}_{\text{adam}}) \approx 0, \quad L_D^{0-1}(\mathbf{W}_{\text{adam}}) \geq 0.5$
- GD can output a point $\mathbf{W}_{\text{gd}}$ in L_2 norm with
  - $L_S(\mathbf{W}_{\text{gd}}) \approx 0, \qquad L_D^{0-1}(\mathbf{W}_{\text{gd}}) \leq 1/\text{poly}(n)$



Adam learns more noisy features than GD (AlexNet on CIFAR-10).

FEATURE LEARNING WITH SIGN GD AND ADAM

Adam/Sign-GD learns noise faster

$$
\begin{aligned}
\langle \boldsymbol{w}_{j,r}^{(t+1)}, j \cdot \boldsymbol{v} \rangle &\leq \langle \boldsymbol{w}_{j,r}^{(t)}, j \cdot \boldsymbol{v} \rangle + \eta \\
\langle \boldsymbol{w}_{y_i,r}^{(t+1)}, \boldsymbol{\xi}_i \rangle &\approx \langle \boldsymbol{w}_{y_i,r}^{(t)}, \boldsymbol{\xi}_i \rangle + \eta s \sigma_\xi
\end{aligned}
$$

noise dominates as $s\sigma_\xi \gg 1$

GD learns signal faster

$$
\begin{aligned}
\langle \boldsymbol{w}_{j,r}^{(t+1)}, j \cdot \boldsymbol{v} \rangle &\geq \langle \boldsymbol{w}_{j,r}^{(t)}, j \cdot \boldsymbol{v} \rangle + \eta \langle \boldsymbol{w}_{j,r}^{(t)}, j \cdot \boldsymbol{v} \rangle^{q-1} \\
\langle \boldsymbol{w}_{j,r}^{(t+1)}, \boldsymbol{\xi}_i \rangle &\leq \langle \boldsymbol{w}_{j,r}^{(t)}, \boldsymbol{\xi}_i \rangle + \eta s \sigma_\xi^2 / n \langle \boldsymbol{w}_{j,r}^{(t)}, \boldsymbol{\xi}_i \rangle^{q-1}
\end{aligned}
$$

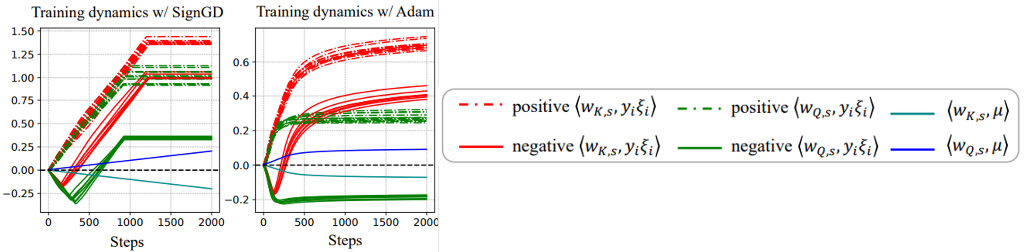
signal dominates as $s\sigma_\xi^2/n \ll 1$

For a two-layer transformer: a similar result holds

Sign GD converges fast but generalize poorly (Li et al., 2025)

There exists T such that

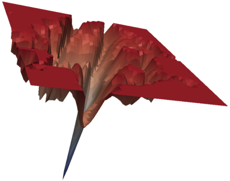
- Training converges but test loss remains large: $L_S^{(T)}(\mathbf{W}^{(T)}) \le \epsilon$, and $L_D^{(T)}(\mathbf{W}^{(T)}) = \Theta(1)$.
- Value, query and key matrices memorizes noise.



FEATURE LEARNING UNDER DIFFERENT OPTIMIZERS (SAM)

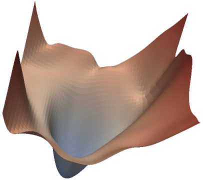
For deep learning, loss landscape is highly nonconvex.

- Minimum found by GD (sharp)



$$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} - \eta \nabla_{\boldsymbol{w}} L(\boldsymbol{w}^{(t)})$$

- Minimum found by SAM (flat)



$$\boldsymbol{g}^{(t)} = \nabla_{\boldsymbol{w}} L \left(\boldsymbol{w}^{(t)} + \tau \frac{\nabla_{\boldsymbol{w}} L(\boldsymbol{w}^{(t)})}{\|\nabla_{\boldsymbol{w}} L(\boldsymbol{w}^{(t)})\|} \right)$$

$$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} - \eta \boldsymbol{g}^{(t)}$$

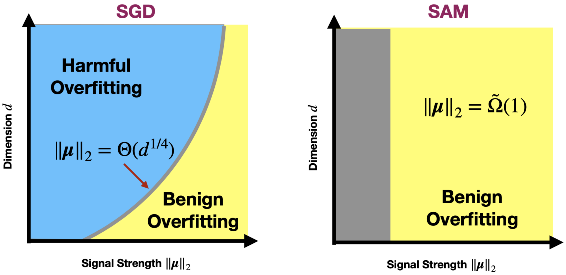
Image Credit: Foret et al. 2021 "Sharpness-aware minimization for efficiently improving generalization"

Benign Overfitting of SAM (Chen et al., 2023)

Under Signal-noise data model $\boldsymbol{x} = [y\boldsymbol{\mu}, \boldsymbol{\xi}]$ and two-layer CNN model with $\sigma = \mathrm{ReLU}$, suppose $\|\boldsymbol{\mu}\| = \widetilde{\Omega}(1)$,[a] then neural network first trained with SAM, then with SGD can find $\mathbf{W}^{(T)}$ with

- small training loss $L_S(\mathbf{W}^{(T)}) \approx 0$ and small test error $L_D^{0-1}(\mathbf{W}^{(T)}) \approx 0$.

[a] this is milder compared to GD Kou et al. (2023), requiring $\|\boldsymbol{\mu}\|^4 = \widetilde{\Omega}(d/n)$.



FEATURE LEARNING UNDER DIFFERENT OPTIMIZERS (LABEL NOISE SGD)

Label Noise SGD: A simple regularization by introducing randomness to labels during training.

For each step t and sample $(\boldsymbol{x}_i, y_i)$:

1. Sample a random variable $\epsilon_i^{(t)}$:

$$\epsilon_i^{(t)} = \begin{cases} 1 & \text{prob } 1-p \quad \text{(Keep)} \\ -1 & \text{prob } p \quad \text{(Flip)} \end{cases}$$

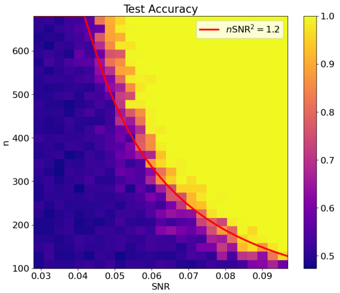
2. Update weights using the noisy gradient:

$$\boldsymbol{w}^{(t+1)} \leftarrow \boldsymbol{w}^{(t)} - \eta \frac{1}{n} \sum_{i=1}^{n} \nabla \ell(\epsilon_i^{(t)} y_i, f(\boldsymbol{x}_i))$$

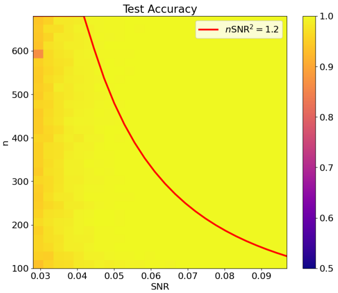
Improved Generalization of Label Noise GD (Huang et al., 2025)

Under Signal-noise data model $\boldsymbol{x} = [y\boldsymbol{\mu}, \boldsymbol{\xi}]$ and two-layer CNN model with $\sigma = \mathrm{ReLU}^2$, then neural network trained with Label Noise GD can find $\mathbf{W}^{(T)}$ with

- constant training loss $L_S(\mathbf{W}^{(T)}) = \Theta(1)$ and small test error $L_D^{0-1}(\mathbf{W}^{(T)}) \approx 0$.



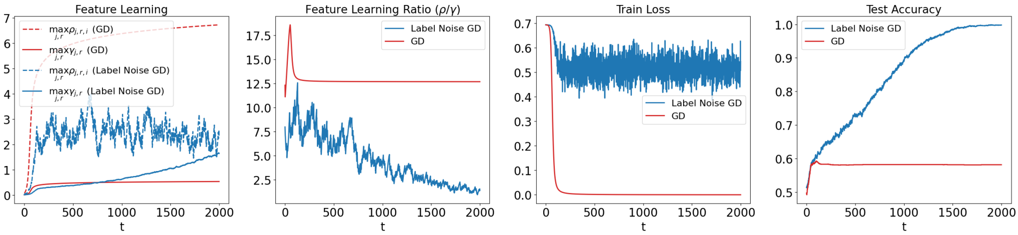
Fails in Low SNR



Robust across regimes

FEATURE LEARNING UNDER DIFFERENT OPTIMIZERS (LABEL NOISE SGD)

Component	Standard GD	Label Noise GD
Signal ($\gamma^{(t)}$)	Grows until loss ≈ 0	Grows exponentially (Stage II)
Noise ($\rho^{(t)}$)	**Dominate & Unbounded**	**Suppressed & Bounded**



OTHER RELEVANT WORKS [1]

Jelassi & Li. 2022. Towards understanding how momentum improves generalization in deep learning. *International Conference on Machine Learning (ICML 2022)*.

Chen et al. 2022. Towards understanding the mixture-of-experts layer in deep learning. *Advances in Neural Information Processing Systems (NeurIPS 2022).*

Zou et al. 2023. The benefits of mixup for feature learning. *International Conference on Machine Learning (ICML 2023).*

Chen et al. 2023. Understanding and improving feature learning for out-of-distribution generalization. *Advances in Neural Information Processing Systems (NeurIPS 2023).*

Pan et al. 2024. Federated learning from vision-language foundation models: Theoretical analysis and method. *Advances in Neural Information Processing Systems (NeurIPS 2024).*

Oh & Yun. 2024. Provable benefit of cutout and cutmix for feature learning. *Advances in Neural Information Processing Systems (NeurIPS 2024).*

OTHER RELEVANT WORKS [2]

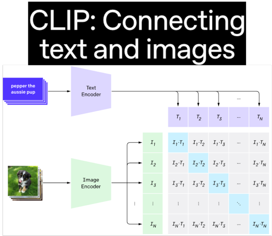
Huang et al. 2024. Understanding convergence and generalization in federated learning through feature learning theory. *International Conference on Learning Representations (ICLR 2024).*

Lu et al. 2024. Benign Oscillation of Stochastic Gradient Descent with Large Learning Rate. *International Conference on Learning Representations (ICLR 2024).*

Oh et al. 2025. From linear to nonlinear: Provable weak-to-strong generalization through feature learning. *Advances in Neural Information Processing Systems (NeurIPS 2025).*

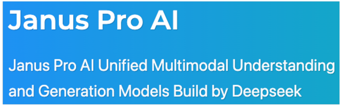
Feature Learning in Foundation Generative Models

FEATURE LEARNING IN FOUNDATION GENERATIVE MODELS



CLIP: Contrastive Pretraining

Diffusion Model



Multimodal LLM

Reasoning LLM

Image credit: https://openai.com/index/clip/, https://stability.ai/stable-image, https://janusai.pro/, https://chat-deep.ai/

MULTI-MODEL CONTRASTIVE LEARNING

Contrastive learning

Draw similar objects (positive) closer. Repel dissimilar objects (negative)

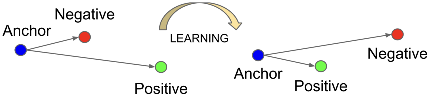


Image credit: Schroff et al. 2015.

MULTI-MODEL CONTRASTIVE LEARNING

Contrastive learning

Draw similar objects (positive) closer. Repel dissimilar objects (negative)

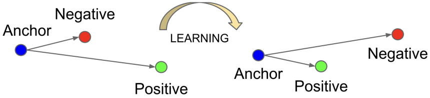
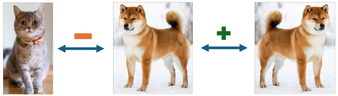
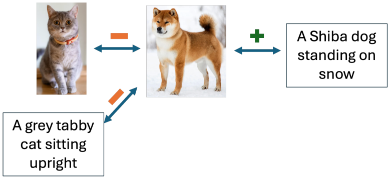


Image credit: Schroff et al. 2015.

Single-Modal Contrastive Learning



Multi-Modal Contrastive Learning



- *Single-Modal*: positive pairs from data augmentation
- *Multi-Modal*: positive pairs from other modalities

WHAT IS THE DIFFERENCE?

MULTI-MODEL CONTRASTIVE LEARNING [1]

Multi-modal Signal-Noise Model (Huang et al., 2024)

- Two modalities:

  $\boldsymbol{x} = [\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}] = [y\boldsymbol{\mu}, \boldsymbol{\xi}]$, $\widetilde{\boldsymbol{x}} = [\widetilde{\boldsymbol{x}}^{(1)}, \widetilde{\boldsymbol{x}}^{(2)}] = [y\widetilde{\boldsymbol{\mu}}, \widetilde{\boldsymbol{\xi}}]$ (label sharing).

- Nonlinear Embedding:

  Let $\mathrm{Emb}(\boldsymbol{x}) = \sigma(\mathbf{W}\boldsymbol{x})$ and $\mathrm{Emb}(\widetilde{\boldsymbol{x}}) = \sigma(\widetilde{\mathbf{W}}\widetilde{\boldsymbol{x}})$ where $\sigma = \mathrm{ReLU}$.

- Data Augmentation:

  $\widehat{\boldsymbol{x}} = [\widehat{\boldsymbol{x}}^{(1)}, \widehat{\boldsymbol{x}}^{(2)}] = [y\boldsymbol{\mu}, \boldsymbol{\xi} + \boldsymbol{\epsilon}], \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma_\epsilon^2 \mathbf{I})$

- Patch-wise Similarity:

  $$\mathrm{Sim}(\boldsymbol{x}, \boldsymbol{x}') = \left\langle \mathrm{Emb}(\boldsymbol{x}^{(1)}), \mathrm{Emb}(\boldsymbol{x}'^{(1)}) \right\rangle + \left\langle \mathrm{Emb}(\boldsymbol{x}^{(2)}), \mathrm{Emb}(\boldsymbol{x}'^{(2)}) \right\rangle$$

MULTI-MODEL CONTRASTIVE LEARNING [2]

- Contrastive Loss:

$$\ell\left(\boldsymbol{x}, \boldsymbol{x}^{+}, \{\boldsymbol{x}_j^{-}\}_{j=1}^{M}\right) = -\log\left(\frac{e^{\mathrm{Sim}(\boldsymbol{x}_i, \widehat{\boldsymbol{x}}_i)/\tau}}{e^{\mathrm{Sim}(\boldsymbol{x}, \widehat{\boldsymbol{x}}')/\tau} + \sum_{j\neq i}^{M} e^{\mathrm{Sim}(\boldsymbol{x}, \widehat{\boldsymbol{x}}')/\tau}}\right)$$

  - Single-Modal:

$$L = \frac{1}{n}\sum_{i=1}^{n} \ell\left(\boldsymbol{x}_i, \widehat{\boldsymbol{x}}_i, \{\boldsymbol{x}_j\}_{j\neq i}^{M}\right)$$

  - Multi-Modal:

$$L = \frac{1}{n}\sum_{i=1}^{n} \ell\left(\boldsymbol{x}_i, \widetilde{\boldsymbol{x}}_i, \{\widetilde{\boldsymbol{x}}_j\}_{j\neq i}^{M}\right)$$

Focus on the feature learning of the first modality, i.e., $\boldsymbol{\mu}$, $\boldsymbol{\xi}_i$.

Multi-modal benefits from *cooperation* between modalities (Huang et al., 2024)

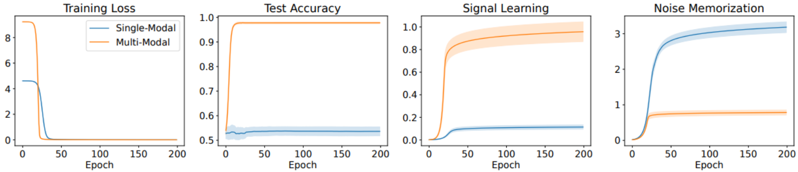
Suppose $n \cdot \mathrm{SNR}^2 = \Theta(1)$ and $\|\widetilde{\boldsymbol{\mu}}\| = C_\mu \|\boldsymbol{\mu}\| > \|\boldsymbol{\mu}\|$.

- Single-Modal: memorize noise (data augmentation does not change SNR)

$$\langle \boldsymbol{w}_r^{(T)}, \boldsymbol{\mu} \rangle \approx 0, \quad \langle \boldsymbol{w}_r^{(T)}, \boldsymbol{\xi}_i \rangle \geq C$$

- Multi-Modal: learn signal ($\|\widetilde{\boldsymbol{\mu}}\| > \|\boldsymbol{\mu}\|$)

$$\langle \boldsymbol{w}_r^{(T)}, \boldsymbol{\mu} \rangle \geq C', \quad \langle \boldsymbol{w}_r^{(T)}, \boldsymbol{\xi}_i \rangle \approx 0$$



DIFFUSION MODEL

$$\mathrm{d}\boldsymbol{x}_t = -\boldsymbol{x}_t \mathrm{d}t + \sqrt{2}\mathrm{d}\boldsymbol{W}_t, \quad \boldsymbol{x}_0 \sim p_0$$

$$\mathrm{d}\boldsymbol{z}_t = -\big(\boldsymbol{z}_t + 2\nabla \log p_t(\boldsymbol{z}_t)\big)\mathrm{d}t + \sqrt{2}\mathrm{d}\overline{\boldsymbol{W}}_t, \quad \boldsymbol{z}_T \sim p_T$$

Diffusion model learns the score function via denoising score matching (Ho et al., 2020)

$$\min_{\mathbf{W}} \mathbb{E}_{\boldsymbol{x}_0 \sim p_0, \boldsymbol{\epsilon} \sim \mathcal{N}(0,\mathbf{I}), t \in [0,T]} \left\| f(\mathbf{W}, \alpha_t \boldsymbol{x}_0 + \beta_t \boldsymbol{\epsilon}, t) - \boldsymbol{\epsilon} \right\|^2$$

DIFFUSION MODEL FEATURE LEARNING

Question: What is the feature learning process of diffusion model? Why do we care?

Diffusion Classifier

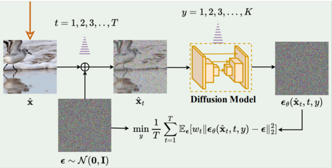


Image credit: Chen, et al. 2024

Semantic Segmentation

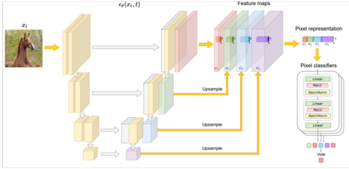


Image credit: Baranchuk, et al. 2022

Image Editing

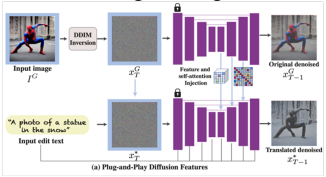


Image credit: Tumanyan et al. 2023

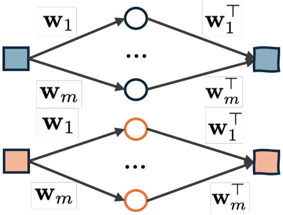
DIFFUSION MODEL FEATURE LEARNING

Han et al. (2025a) compares feature learning of diffusion model with classification models

Diffusion Model

$$\boldsymbol{f}(\mathbf{W}, \boldsymbol{x}) = \left[\boldsymbol{f}_1(\mathbf{W}, \boldsymbol{x}^{(1)}), \boldsymbol{f}_2(\mathbf{W}, \boldsymbol{x}^{(2)})\right]^\top,$$

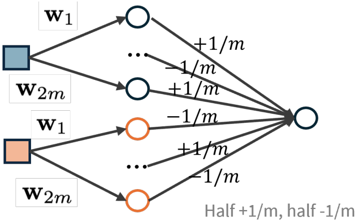
$$\boldsymbol{f}_p(\mathbf{W}, \boldsymbol{x}^{(p)}) = \frac{1}{\sqrt{m}} \sum_{r=1}^{m} \sigma(\langle \boldsymbol{w}_r, \boldsymbol{x}^{(p)} \rangle) \boldsymbol{w}_r$$



Classification Model

$$f(\mathbf{W}, \boldsymbol{x}) = F_1(\mathbf{W}_1, \boldsymbol{x}) - F_{-1}(\mathbf{W}_{-1}, \boldsymbol{x}),$$

$$F_j(\mathbf{W}, \boldsymbol{x}) = \frac{1}{m} \sum_{r=1}^{m} \sum_{p=1,2} \sigma(\langle \boldsymbol{w}_{j,r}, \boldsymbol{x}^{(p)} \rangle)$$



DIFFUSION MODEL FEATURE LEARNING

(Han et al., 2025a, Theorem 3.1 & 3.2)

Diffusion Model learns *balanced* features

- There exists a stationary point $\mathbf{W}^*$ such that

$$|\langle \boldsymbol{w}_r^*, \boldsymbol{\mu}\rangle| / |\langle \boldsymbol{w}_r^*, \boldsymbol{\xi}\rangle| = \Theta(n \cdot \mathrm{SNR}^2)$$

Classification learns *dominate* features

- There exists $\mathbf{W}^*$ with $L_S(\mathbf{W}^*) \approx 0$:
  - When $n \cdot \mathrm{SNR}^2 \geq \overline{C}$, then $|\langle \boldsymbol{w}_r^*, \boldsymbol{\mu}\rangle| \geq C, \quad |\langle \boldsymbol{w}_r^*, \boldsymbol{\xi}_i\rangle| \approx 0$ (Signal dominates)
  - When $n \cdot \mathrm{SNR}^2 \leq \underline{C}$, then $|\langle \boldsymbol{w}_r^*, \boldsymbol{\mu}\rangle| \approx 0, \quad |\langle \boldsymbol{w}_r^*, \boldsymbol{\xi}_i\rangle| \geq C'$ (Noise dominates)

IN-CONTEXT LEARNING

In-context learning is the ability of LLMs that learn new rules with few examples.

Fill in the blank with one word: Apple - red, Watermelon - ____

Apple - red, Watermelon - **green**

Fill in the blank: 311 - 5, 4569 - 24, 12 - ___

311 - 5, 4569 - 24, 12 - **3**

Question: Can we understand in-context learning from feature learning?

IN-CONTEXT LEARNING FEATURE LEARNING

Bu et al. (2024): Each prompt contains a shared concept/task, with the input

$$\mathbf{H} = \begin{pmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_L & \boldsymbol{x}_q \\ \boldsymbol{y}_1 & \boldsymbol{y}_2 & \cdots & \boldsymbol{y}_L & \mathbf{0} \end{pmatrix}, \qquad \text{Goal: predict } y_q$$

Each concept k encodes binary semantics $y = \pm 1$:

$$\boldsymbol{x}_\ell \in \{\boldsymbol{a}_k + y\boldsymbol{b}_k\}, \qquad \boldsymbol{y}_\ell \in \{\boldsymbol{c}_k + y\boldsymbol{d}_k\}$$

Training data $(\mathbf{H}_n, y_n)_{n=1}^N$:

- Sample $k \in [K]$ and $y \in \{\pm 1\}$
- Construct query $\boldsymbol{x}_q = \boldsymbol{a}_k + y\boldsymbol{b}_k + \boldsymbol{\xi}$, $\boldsymbol{y}_q = \boldsymbol{c}_k + y\boldsymbol{d}_k + \boldsymbol{\xi}'$
- Sample prompt examples: $y_\ell \in \{\pm 1\}$, $\boldsymbol{x}_\ell = \boldsymbol{a}_k + y_\ell\boldsymbol{b}_k + \boldsymbol{\xi}_\ell$, $\boldsymbol{y}_\ell = \boldsymbol{c}_k + y_\ell\boldsymbol{d}_k + \boldsymbol{\xi}'_\ell$, $\ell \in [L]$

IN-CONTEXT LEARNING FEATURE LEARNING [1]

Suppose we train a two-layer transformer on $(\mathbf{H}_n, y_n)$ with expected *cross entropy loss*

$$f(\mathbf{H}) = \boldsymbol{r}^\top \mathrm{ReLU}\left(\mathbf{W}_O \,\mathrm{attn}(\mathbf{H})\right), \quad \mathrm{attn}(\mathbf{H}) = \sum_{\ell=1}^{L} \mathbf{W}_V \boldsymbol{h}_\ell \,\mathrm{smax}\left(\boldsymbol{h}_\ell^\top \mathbf{W}_K^\top \mathbf{W}_Q \boldsymbol{h}_q\right)$$

with the following parameterization

$$\mathbf{W}_Q = \begin{pmatrix} \mathbf{W}_Q^x & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \quad \mathbf{W}_K = \begin{pmatrix} \mathbf{W}_K^x & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \quad \mathbf{W}_V = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}, \quad \mathbf{W}_O = \begin{pmatrix} \mathbf{0} & \boldsymbol{w}_O^y \end{pmatrix}$$

Attention only attends to demo inputs and output only depends on demo output.

IN-CONTEXT LEARNING FEATURE LEARNING [2]

Transformer learns *concepts* and *semantics* for in-context learning (Bu et al., 2024)

Upon convergence

- $\mathbf{W}_Q, \mathbf{W}_K$ learn *semantics* rather than concept:

$$\boldsymbol{a}_k^\top \mathbf{W}_Q^x \boldsymbol{a}_k \approx 0, \quad \boldsymbol{b}_k^\top \mathbf{W}_Q^x \boldsymbol{b}_k = \Omega(1),$$
$$\boldsymbol{a}_k^\top \mathbf{W}_K^x \boldsymbol{a}_k \approx 0, \quad \boldsymbol{b}_k^\top \mathbf{W}_K^x \boldsymbol{b}_k = \Omega(1),$$

- $\mathbf{W}_O$ learn *both concept and semantics*

$$\langle \boldsymbol{w}_O^y, \boldsymbol{c}_k \rangle, \langle \boldsymbol{w}_O^y, \boldsymbol{d}_k \rangle = \Omega(1)$$

$\implies$ this allows to leverage label contains in semantics of $\boldsymbol{x}_q$ for output prediction

NONPARAMETRIC IN-CONTEXT FEATURE LEARNING

Each task k is defined via a task function F_k° (Kim et al., 2024; Kim and Suzuki, 2024)

$$F_k^\circ(\boldsymbol{x}) = \boldsymbol{\beta}_k^\top \boldsymbol{f}^\circ(\boldsymbol{x})$$

where $\boldsymbol{\beta}_k$ is (linear) task-specific and $\boldsymbol{f}^\circ(\boldsymbol{x})$ is (nonlinear) task-common features.

Key idea:

- *Pretraining*: learn $\boldsymbol{f}^\circ$
- *In-context*: adapt to $\boldsymbol{\beta}_k$

NONPARAMETRIC IN-CONTEXT FEATURE LEARNING

Given pretraining-data (K tasks)

$$\left\{ \begin{pmatrix} \boldsymbol{x}_{1,k} & \cdots & \boldsymbol{x}_{L,k} & \boldsymbol{x}_{q,k} \\ y_{1,k} & \cdots & y_{L,k} & 0 \end{pmatrix}, y_{q,k} \right\}_{k=1}^{K}$$

Kim and Suzuki (2024) considers linear transformer

$$\frac{1}{L} \sum_{\ell=1}^{L} y_{\ell,k} \boldsymbol{h}_\mu(\boldsymbol{x}_{\ell,k})^\top \boldsymbol{\Gamma} \boldsymbol{h}_\mu(\boldsymbol{x}_{q,k}) \xrightarrow{\text{predict}} y_{q,k}$$

with a mean-field neural network as feature embedding (*infinite limit of two-layer MLP*):

$$\underbrace{\boldsymbol{h}_{\theta_m}(\boldsymbol{x}) = \frac{1}{m} \sum_{r=1}^{m} \boldsymbol{a}_r \sigma(\boldsymbol{w}_r^\top \boldsymbol{x})}_{\text{two-layer MLP}} \xrightarrow{\text{as } m \to \infty} \underbrace{\boldsymbol{h}_\mu(\boldsymbol{x}) = \int \boldsymbol{a} \sigma(\boldsymbol{w}^\top \boldsymbol{x}) \mathrm{d}\mu(\boldsymbol{a}, \boldsymbol{w})}_{\text{mean-field limit}}$$

NONPARAMETRIC IN-CONTEXT FEATURE LEARNING

Minimize expected ICL risk ($K \to \infty, L \to \infty$) w.r.t. μ and $\mathbf{\Gamma}$

$$\mathcal{L}(\mu, \mathbf{\Gamma}) = \mathbb{E}_{\boldsymbol{x}_q}\left[\|\boldsymbol{f}^\circ(\boldsymbol{x}_q) - \mathbb{E}_{\boldsymbol{x}}[\boldsymbol{f}^\circ(\boldsymbol{x})\boldsymbol{h}_\mu(\boldsymbol{x})^\top]\mathbf{\Gamma}\boldsymbol{h}_\mu(\boldsymbol{x}_q)\|^2\right]$$

Key idea: pretraining to $\mathcal{L} = 0$, so that for unseen task $\boldsymbol{\beta}_{\text{new}}$,

$$\tilde{y}_q = \mathbb{E}_{\boldsymbol{x}}[\boldsymbol{\beta}_{\text{new}}^\top \boldsymbol{f}^\circ(\boldsymbol{x})\boldsymbol{h}_\mu(\boldsymbol{x})^\top]\mathbf{\Gamma}\boldsymbol{h}_\mu(\tilde{\boldsymbol{x}}_q) = \boldsymbol{\beta}_{\text{new}}^\top \boldsymbol{f}^\circ(\tilde{\boldsymbol{x}}_q)$$

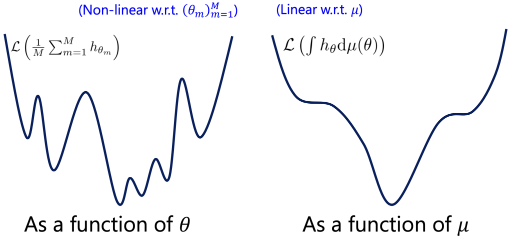


Image credit: Suzuki 2024.

NONPARAMETRIC IN-CONTEXT FEATURE LEARNING

Feature learning under two time-scale dynamics ($\mathbf{\Gamma}$ converges first)

$$\mathcal{L}(\mu) = \min_{\mathbf{\Gamma}} \mathcal{L}(\mu, \mathbf{\Gamma}) = \mathbb{E}_{\boldsymbol{x}_q} \left[\| \boldsymbol{f}^\circ(\boldsymbol{x}_q) - \mathbf{\Sigma}_{\mu^\circ,\mu} \mathbf{\Sigma}_{\mu,\mu}^{-1} \boldsymbol{h}_\mu(\boldsymbol{x}_q) \|^2 \right]$$

where $\mathbf{\Sigma}_{\mu,\nu} = \mathbb{E}_{\boldsymbol{x}}[\boldsymbol{h}_\mu(\boldsymbol{x}) \boldsymbol{h}_\nu^\top(\boldsymbol{x})]$ is the feature covariance, and μ° satisfies $\boldsymbol{h}_{\mu^\circ} = \boldsymbol{f}^\circ$.

Wasserstein gradient flow escapes strict saddles and converges to global minimum (Kim and Suzuki, 2024):

- Nonlinear feature learning: $\boldsymbol{h}_\mu \to \boldsymbol{R}\boldsymbol{f}^\circ$
  (for some invertible matrix $\boldsymbol{R}$ with bounded norm).

OTHER RELEVANT WORKS [1]

Li et al. 2024. How do nonlinear transformers learn and generalize in in-context learning? *International Conference on Machine Learning (ICML 2024).*

Nichani et al. 2024. How transformers learn causal structure with gradient descent. *International Conference on Machine Learning (ICML 2024).*

Chen et al. 2024. Unveiling induction heads: Provable training dynamics and feature learning in transformers. *Advances in Neural Information Processing Systems (NeurIPS 2024).*

Oko et al. 2024. Pretrained transformer efficiently learns low-dimensional target functions in-context. *Advances in Neural Information Processing Systems (NeurIPS 2024).*

Bu et al. 2025. Provable In-Context Vector Arithmetic via Retrieving Task Concepts. *International Conference on Machine Learning (ICML 2025).*

Nishikawa et al. 2025. Nonlinear transformers can perform inference-time feature learning. *International Conference on Machine Learning (ICML 2025).*

OTHER RELEVANT WORKS [2]

Yang et al. 2025. Multi-head Transformers Provably Learn Symbolic Multi-step Reasoning via Gradient Descent. *Advances in Neural Information Processing Systems (NeurIPS 2025).*

Conclusion and Outlook

CONCLUSION AND OUTLOOK

- **Feature learning** underlies the *success of deep learning* and provides a *theoretical* framework for understanding, controlling and improving deep learning
  - Benign overfitting (CNN, Transformer)
  - Training strategies (Adam, Sign-GD, SAM, Label noise)
  - Foundation models (contrastive pre-training, diffusion models, in-context learning)
  - and many more

Understanding: Unbox the black-box to study internal representation

Controlling: Manipulate the latent features for controlled model output

Improving: Leverage learned features for model safety, privacy, and robustness.

THANK YOU!

REFERENCES [1]

Bu, D., Huang, W., Han, A., Nitanda, A., Suzuki, T., Zhang, Q., and Wong, H.-S. (2024). Provably transformers harness multi-concept word semantics for efficient in-context learning. *Advances in Neural Information Processing Systems*, 37:63342–63405.

Cao, Y., Chen, Z., Belkin, M., and Gu, Q. (2022). Benign overfitting in two-layer convolutional neural networks. *Advances in Neural Information Processing Systems*, 35:25237–25250.

Chen, Z., Zhang, J., Kou, Y., Chen, X., Hsieh, C.-J., and Gu, Q. (2023). Why does sharpness-aware minimization generalize better than sgd? *Advances in Neural Information Processing Systems*, 36:72325–72376.

Han, A., Huang, W., Cao, Y., and Zou, D. (2025a). On the feature learning in diffusion models. In *The Thirteenth International Conference on Learning Representations*.

REFERENCES [2]

Han, A., Huang, W., Zhou, Z., Niu, G., Chen, W., Yan, J., Takeda, A., and Suzuki, T. (2025b). On the role of label noise in the feature learning process. In *International Conference on Machine Learning*.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851.

Huang, W., Han, A., Chen, Y., Cao, Y., Xu, Z., and Suzuki, T. (2024). On the comparison between multi-modal and single-modal contrastive learning. *Advances in Neural Information Processing Systems*, 37:81549–81605.

Huang, W., Han, A., Song, Y., Chen, Y., Wu, D., Zou, D., and Suzuki, T. (2025). How does label noise gradient descent improve generalization in the low snr regime? *Advances in Neural Information Processing Systems*.

REFERENCES [3]

Jiang, J., Huang, W., Zhang, M., Suzuki, T., and Nie, L. (2024). Unveil benign overfitting for transformer in vision: Training dynamics, convergence, and generalization. *Advances in Neural Information Processing Systems*, 37:135464–135625.

Kim, J., Nakamaki, T., and Suzuki, T. (2024). Transformers are minimax optimal nonparametric in-context learners. *Advances in Neural Information Processing Systems*, 37:106667–106713.

Kim, J. and Suzuki, T. (2024). Transformers learn nonlinear features in context: nonconvex mean-field dynamics on the attention landscape. In *International Conference on Machine Learning*, pages 24527–24561.

Kou, Y., Chen, Z., Chen, Y., and Gu, Q. (2023). Benign overfitting in two-layer relu convolutional neural networks. In *International Conference on Machine Learning*, pages 17615–17659. PMLR.

REFERENCES [4]

Li, B., Huang, W., Han, A., Zhou, Z., Suzuki, T., Zhu, J., and Chen, J. (2025). On the optimization and generalization of two-layer transformers with sign gradient descent. In *International Conference on Learning Representations*.

Zou, D., Cao, Y., Li, Y., and Gu, Q. (2023). Understanding the generalization of adam in learning neural networks with proper regularization. In *The Eleventh International Conference on Learning Representations*.