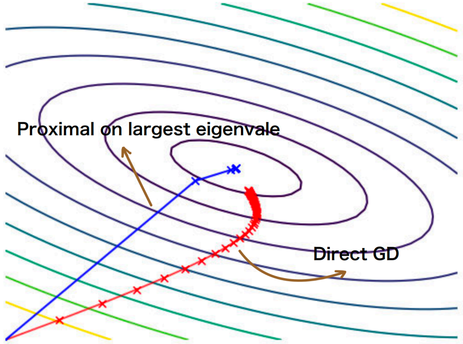
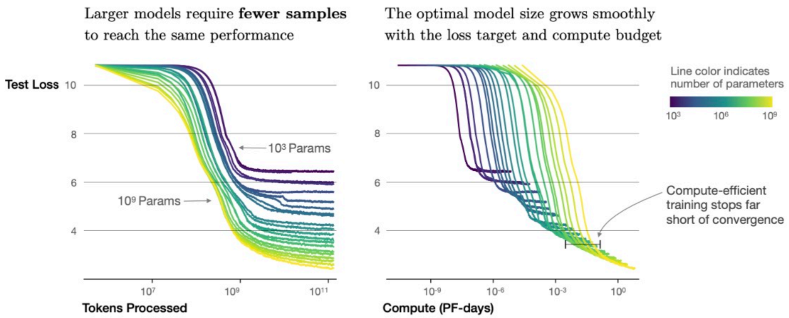
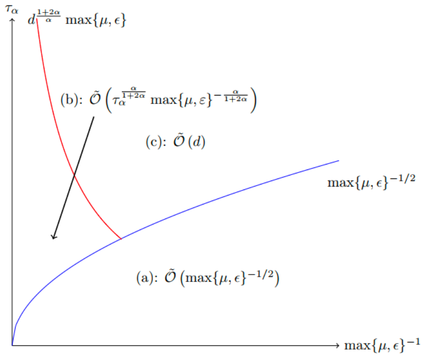
Implicit Regularization of SGD ： Multiple Descent, Emergence, Algorithm Design

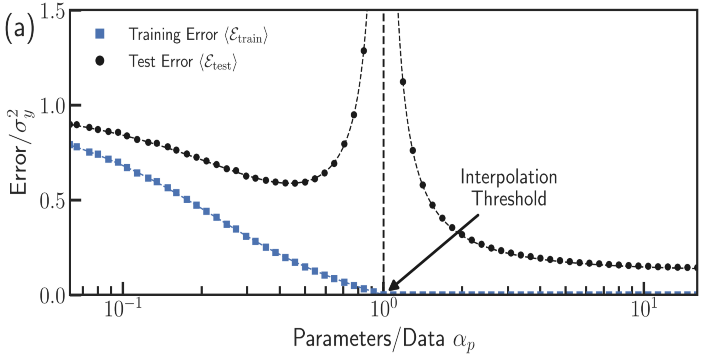
Cong Fang

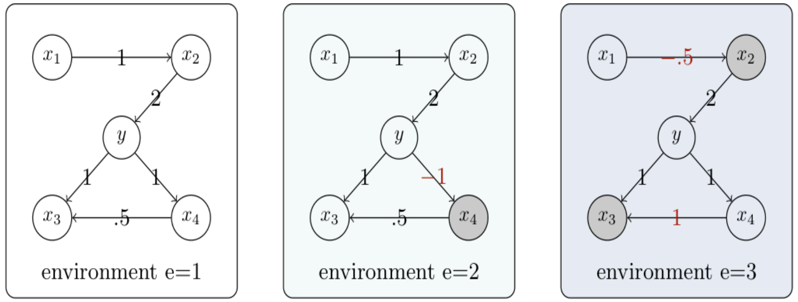
Research Overview











Optimization:

- Algorithm Design (NeurIPS2024, ICLR2025,JMLR2024)
- Improved Analysis (COLT2023)
- Lower Bounds (COLT2023)

Optimization based Generalization：

- Linear/Non-linear Model
- Offline/Stochastic Algorithms
- Data Structure
- Explicit Comparison

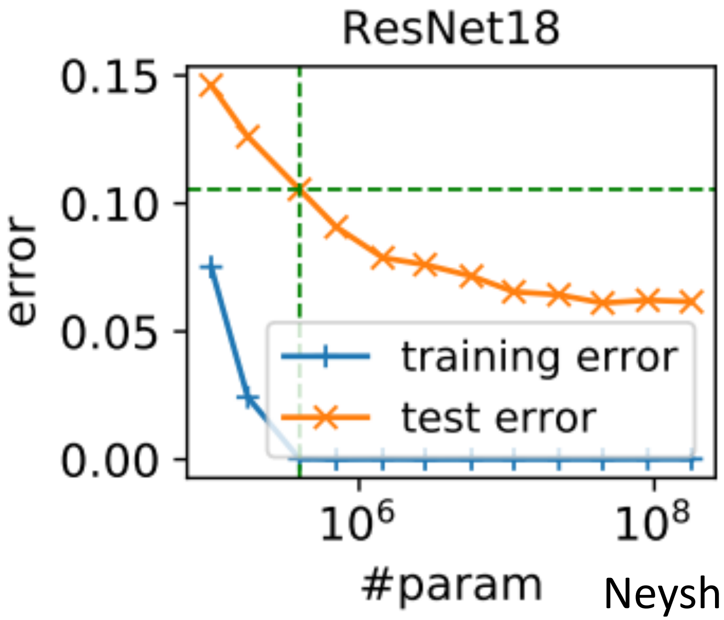
Invariance Based Causal Discovery：

- Linear Model (AoS 2024)
- Non-Linear Model (AoS 2025)
- Computational Complexity
- Convex Relaxation (NeurIPS2024)

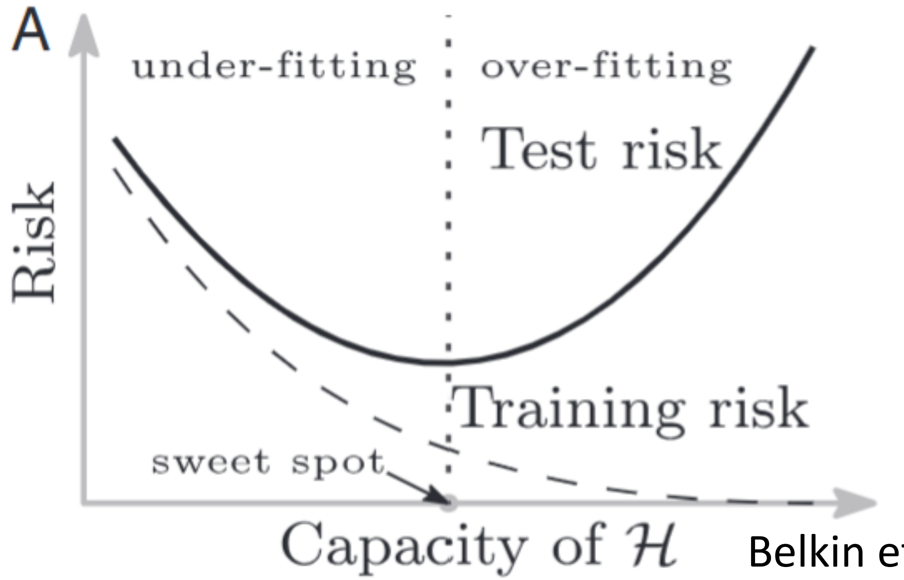
Benign Overffitting Phenomenon

- Modern neural network parameters $\gg$ The number of samples
- Overparameterized models trained with GD/SGD can achieve good generalization.
- Challenging the traditional uniform convergence generalization theory.

Why can good generalization still be achieved?



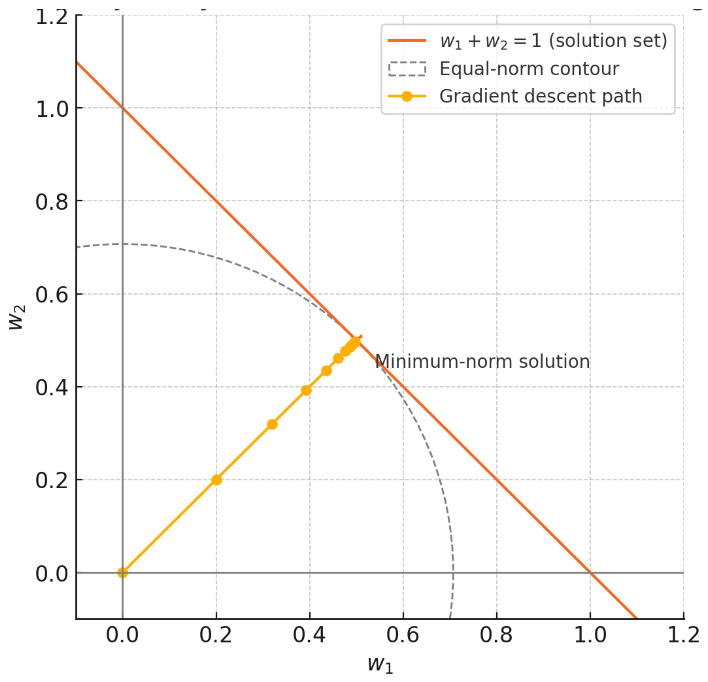
Neyshabur et al. (2019)

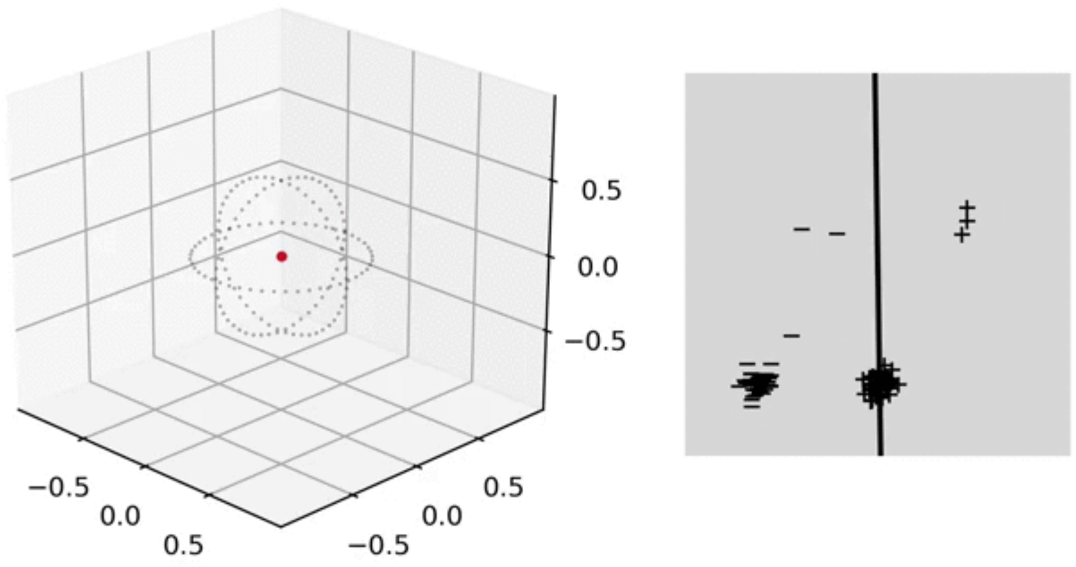


Belkin et al. (2019)

Implicit Regularization

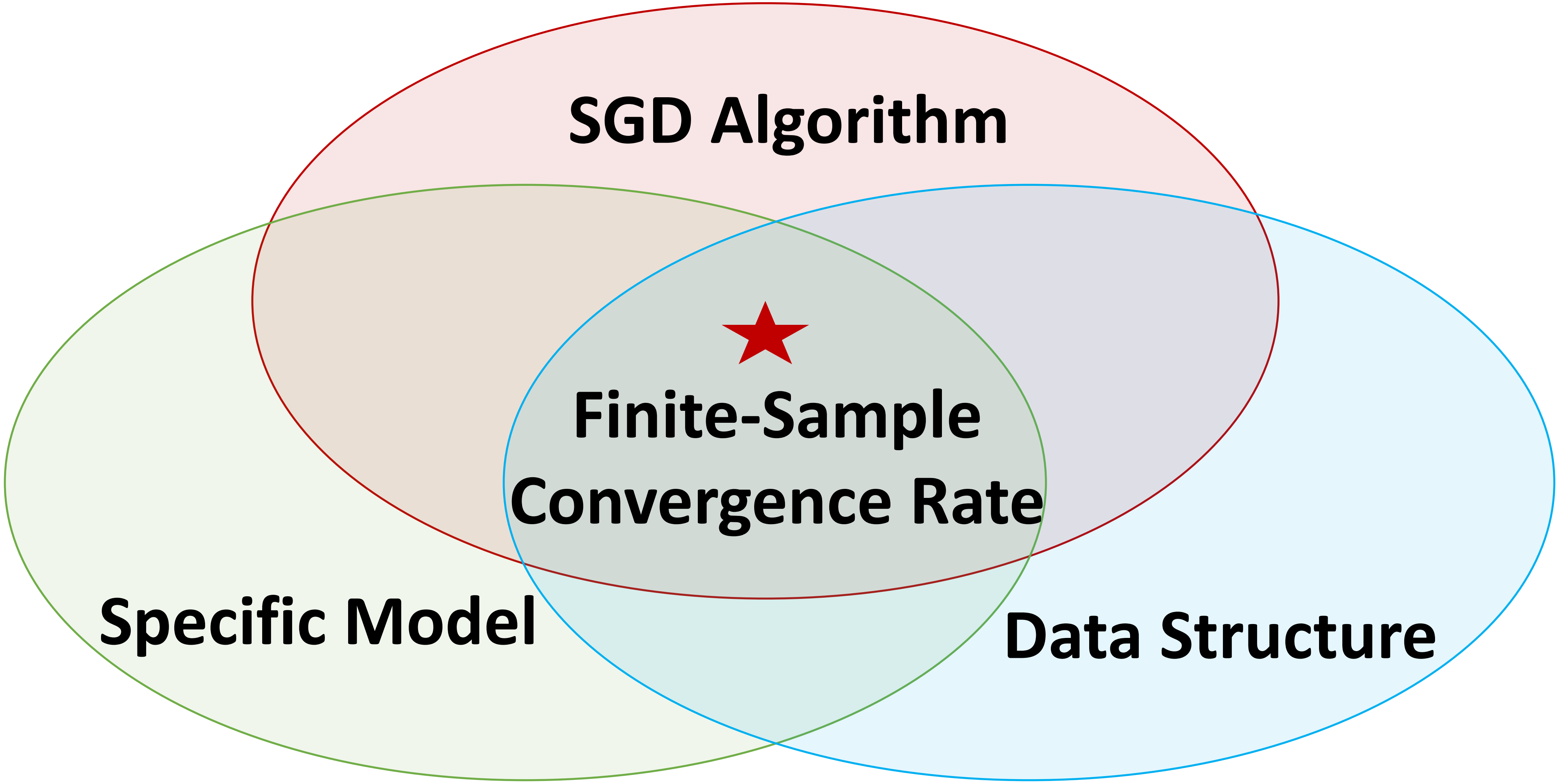
- Implicit Regularization: **Algorithm** guides the model to converge to **solutions with special properties** without explicit regularization.
- **GD** converges to the **max-margin solution** on logistic regression. (Soudry et al., 2018)
- **GD** converges to the **min-norm solution** on linear regressiom. (Belkin et al. , 2019)





Francis Bach’s Blog

SGD's Implicit Regularization



SGD's Implicit Regularization

Model

Linear Model

$f(\mathbf{w}) = \langle \mathbf{w}, \mathbf{x} \rangle$

Non-Linear

Quadratically Parameterized Model

$f(\mathbf{w}) = \left\langle \mathbf{w}^{\odot 2}, \mathbf{x} \right\rangle$

Neural Network

$f(\mathbf{W}, \mathbf{a}) = \frac{1}{m} \sum_{i=1}^{m} a_i \sigma\left(\langle \mathbf{w}_i, \mathbf{x} \rangle\right)$

SGD's Implicit Regularization

- SGD is a practical method in deep learning training.
- SGD introduces anisotropic noise during training.
- The stochastic dynamics of SGD can be directly analyzed
- Obtain generalization convergence rates with respect to sample size n.

Data

In-Distribution

- Low Input Dimension
- High Input Dimension

Out-of-Distribution

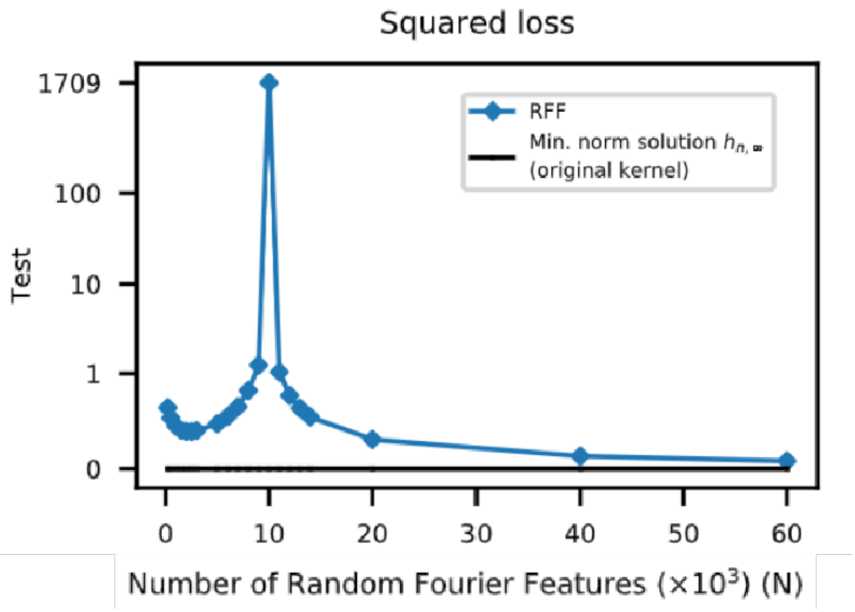
- Covariate Shift

北京大學
PEKING UNIVERSITY

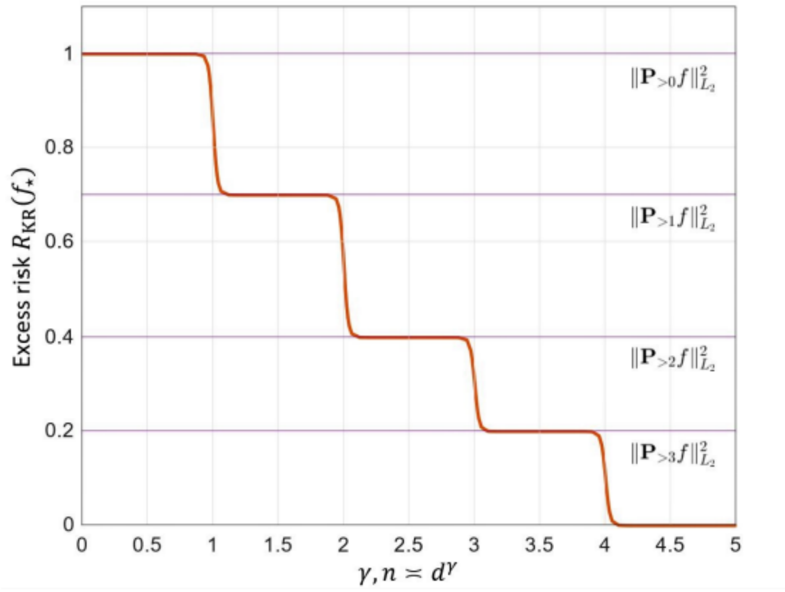
Learning Curves of Stochastic Gradient Descent in Kernel Regression

Background

Double and Multiple Descent



Belkin et al. (2019), PNAS



Ghorbani et al. (2021), AOS

- Current research primarily focuses on the minimum-norm solution and offline algorithms.

How the scale interplay of dimensionality and sample size impacts the generalization performance of SGD?

Settings

Regression in RKHS

$$(\mathbf{x}_i, y_i) \in \mathbb{S}^{d+1} \times \mathbb{R}, \quad i \in [n], \quad \overset{\text{i.i.d.}}{\sim} \quad y = f_*(\mathbf{x}) + \epsilon, \quad \mathbb{E}^2\left[\epsilon|\mathbf{x}\right] \leq \sigma^2.$$

Goal: minimize $\|f - f_*\|^2_{L^2(\mathbb{S}^{d+1}, \text{Unif}(\mathbb{S}^{d+1}))}$.

Kernel: The NTK of ReLU netwok with inputs on $\mathbb{S}^{d+1}$.

Mercer Decomposition: $K(\mathbf{x}, \mathbf{y}) = \sum_{k=0} \lambda_k \sum_{j=1}^{N(d,k)} Y_{k,j}(\mathbf{x}) Y_{k,j}(\mathbf{y})$

Interpolation space: $[\mathcal{H}]^s = \left\{ \sum_{i=1}^{\infty} a_i \lambda_i^{\frac{s}{2}} \phi_i \mid \{a_i\}_{i=1}^{\infty} \in \ell^2 \right\}.$

Consider different n-to-d ratios; source condition $\|f_*\|_{[\mathcal{H}]^s} \leq 1, \quad s > 0.$

Algorithm

SGD for Kernel Regression

$$f_{t+1} = f_t - \eta_t \left(f_t(\mathbf{x}_t) - y_t\right) K_{\mathbf{x}_t}.$$

If $f \equiv 0$, f_{t+1} can be expressed as: $f_{t+1} = \sum_{j=1}^{t} a_j K_{\mathbf{x}_j}$,

$$\text{where } a_0 = 0,\ a_t = -\eta_{t-1} \left(\sum_{j=1}^{t-1} a_j K\left(\mathbf{x}_j, \mathbf{x}_t\right) - y_t \right).$$

Step Size Schedule

Exponentially Decay: Given a total of n iterations, $\eta_t = \dfrac{\eta_0}{2^{\ell-1}}$,

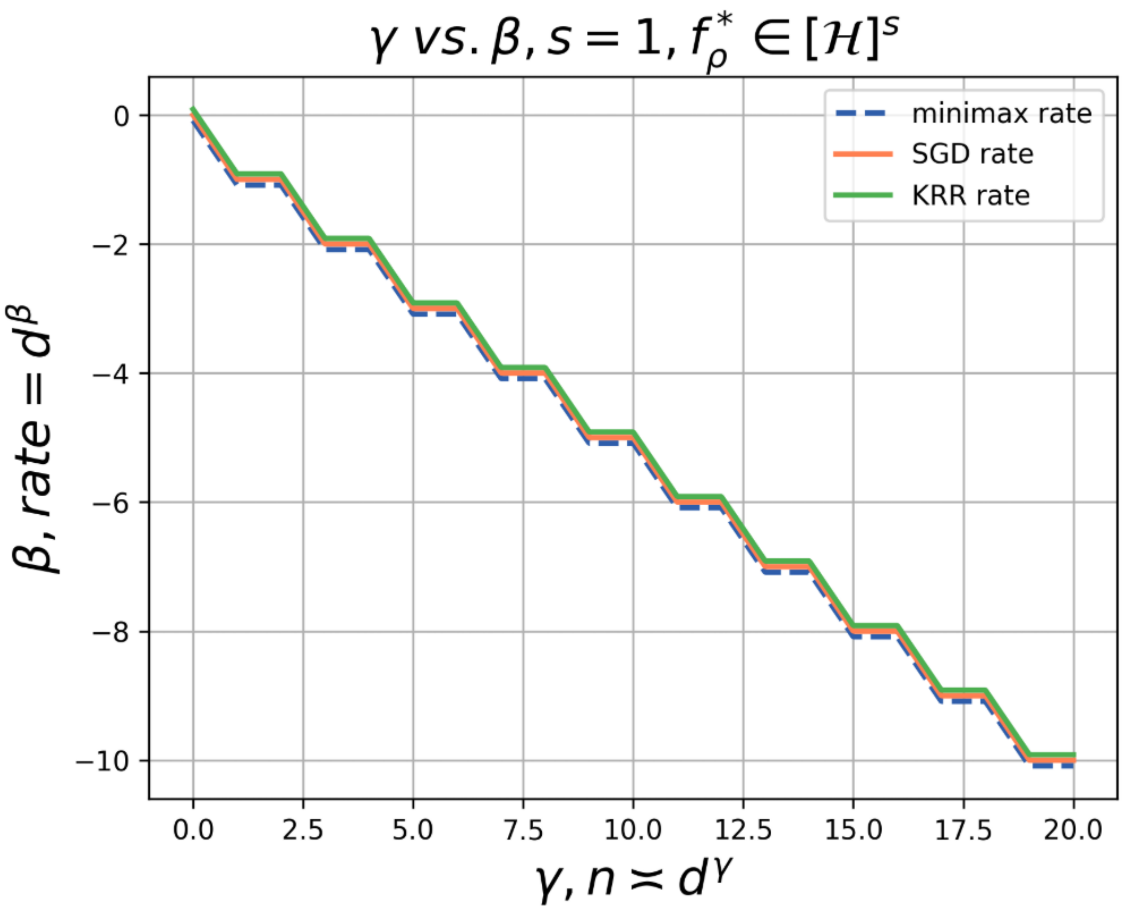
$$\text{if } m(\ell-1)+1 \le t \le m\ell, m = \lceil \frac{n}{\log_2 n} \rceil.\ f_n^{dec} = f_n.$$

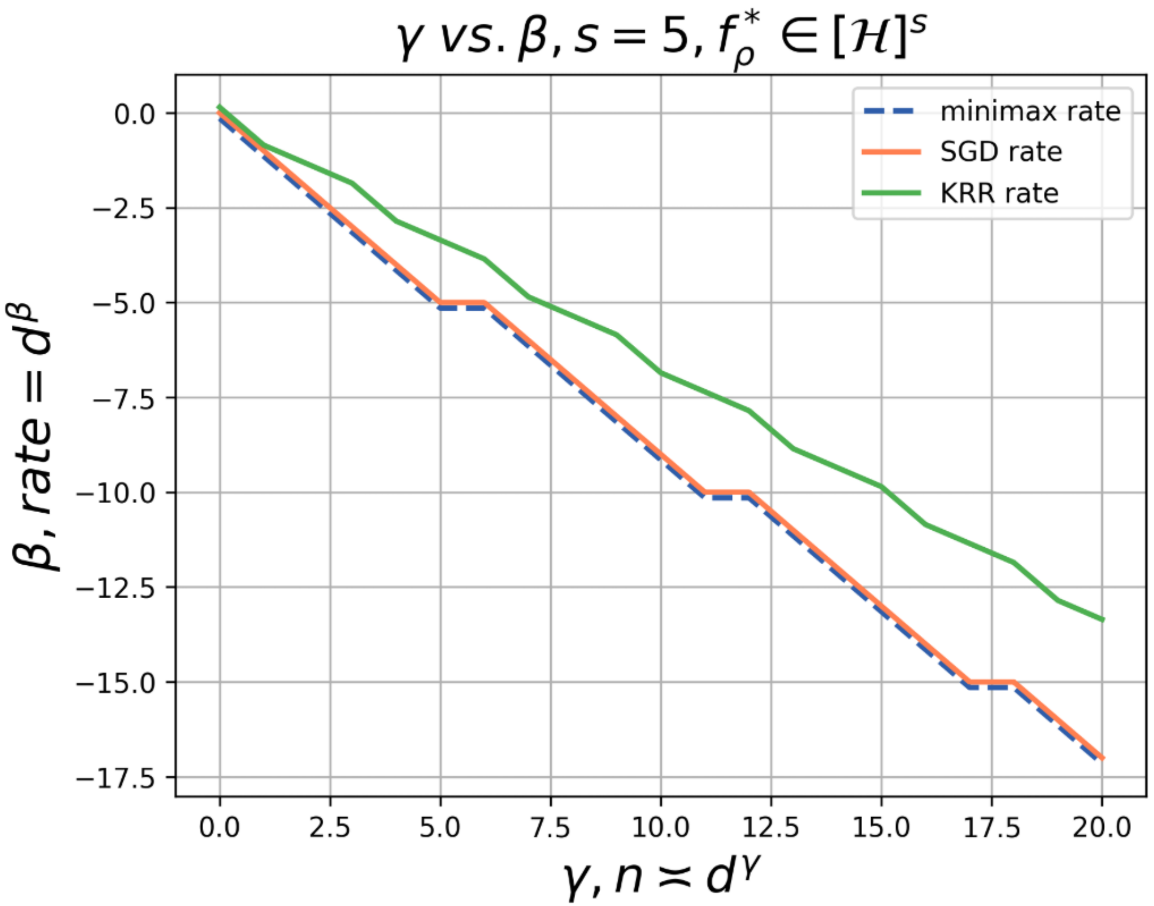
Constant Step Size with Averaged Iterates: $\eta_t \equiv \eta_0$, $f_n^{avg} = \dfrac{1}{n} \sum_{t=0}^{n-1} f_t$.

Convergence Rates in High-Dimensional Settings

When $n \asymp d^{\gamma},\ \|f_*\|_{[\mathcal{H}]^s} \leq 1,$ SGD can achieve optimality for all $s > 0$.

While KRR cannot for $s > 1$.

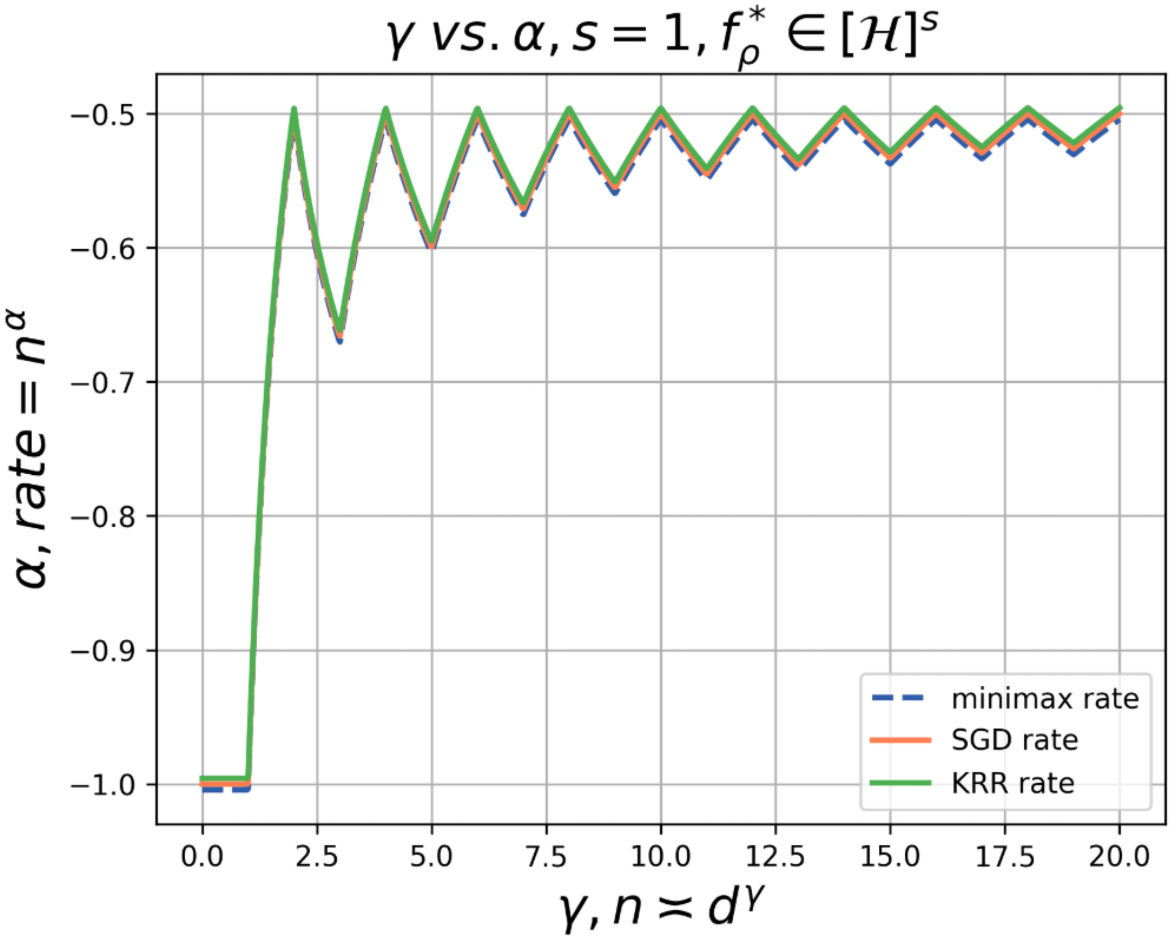


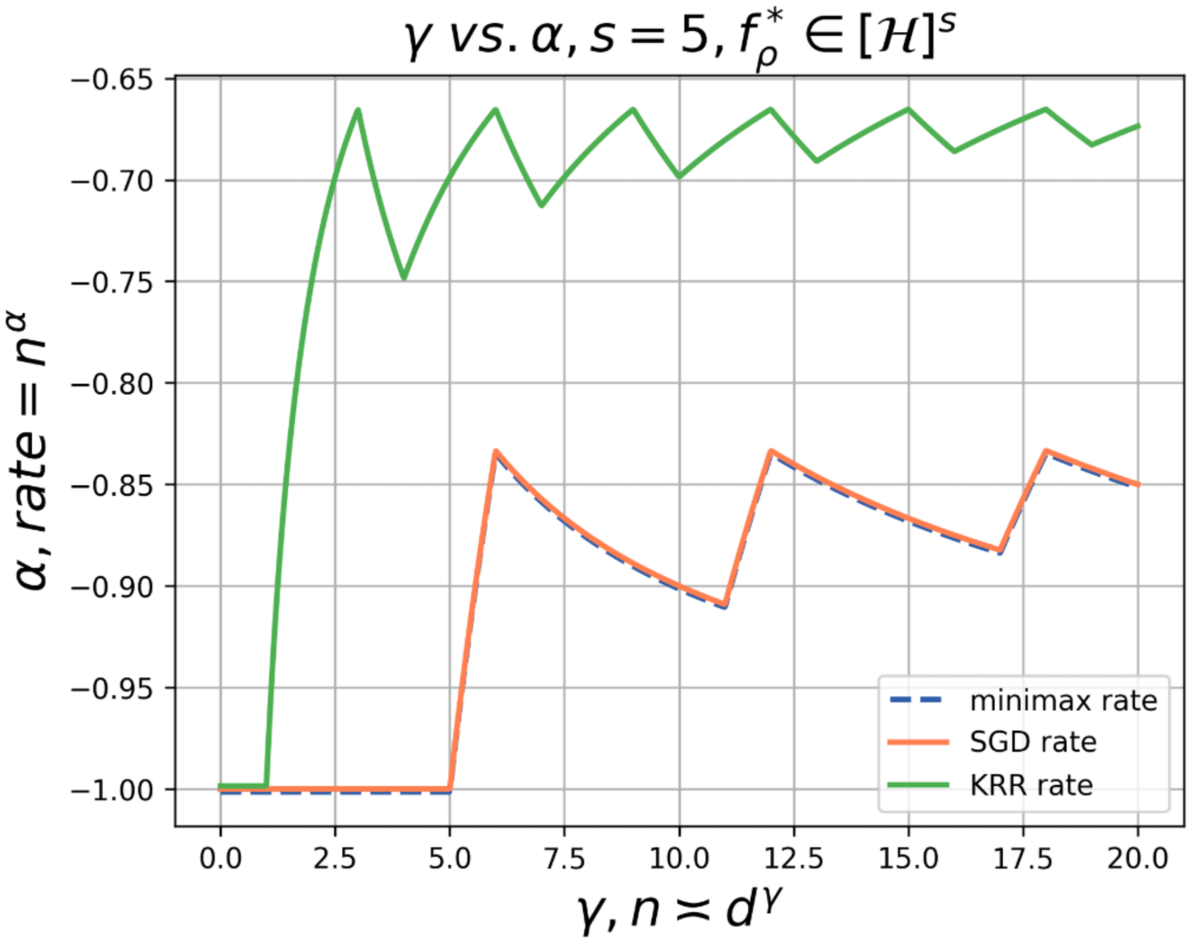


Convergence Rates in High-Dimensional Settings

When $n \asymp d^{\gamma}, \ \|f_*\|_{[\mathcal{H}]^s} \leq 1,$ SGD can achieve optimality for all $s > 0$.

While KRR cannot for $s > 1$.

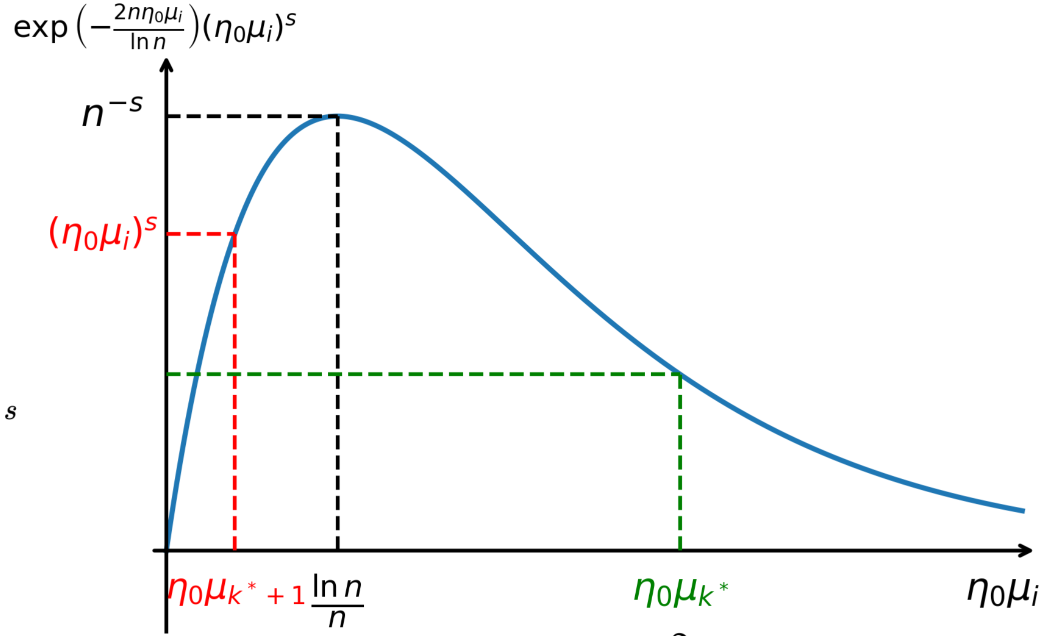




Proof Sketch: General Settings

$$K(\mathbf{x},\mathbf{y}) = \sum_{k=0} \mu_k \phi_k(\mathbf{x}) \phi_k(\mathbf{y}), \quad \mu_1 \geq \mu_2 \geq \cdots, \quad k^* = \max_{i\in\mathbb{N}_+}\{k : \eta_0 \mu_k \geq \frac{\ln n}{n}\}$$

Excess Risk $\lesssim$ Bias + Variance



Bias: $\leq \frac{1}{\eta_0^s}||(\mathbf{I} - \eta_0 \mathbf{\Sigma})^m (\eta_0 \mathbf{\Sigma})^{\frac{s}{2}}||^2 ||f_\rho^*||^2_{[\mathcal{H}]^s}$

$$\lesssim \eta_0^{-s} \max_{i\geq 0} \exp(-2\frac{n\eta_0\mu_i}{\ln n})(\eta_0\mu_i)^s \|f_*\|^2_{[\mathcal{H}]^s}$$

$$\approx \mu_{k^*+1}^s \|f_*\|^2_{[\mathcal{H}]^s}$$

$\approx$Exponentially Converge $(i \leq k^*)$ $\quad$ $\lesssim \lambda_{k^*+1}^s \|f_*\|^2_{[\mathcal{H}]^s} \quad (i > k^*)$

μ_1 $\quad$ μ_{k^*} $\quad$ $\frac{\ln n}{n\eta_0}$ $\quad$ μ_{k^*+1}

Variance: $\lesssim k^* \times \frac{1}{n}$ **+** $\lesssim \sum_{i=k^*+1}^{\infty} n\eta_0^2\mu_i^2$

Proof Sketch: High-Dimensional Settings

$$d \asymp n^{\gamma}, \quad K(\mathbf{x}, \mathbf{y}) = \sum_{k=0} \lambda_k \sum_{j=1}^{N(d,k)} Y_{k,j}(\mathbf{x}) Y_{k,j}(\mathbf{y}).$$

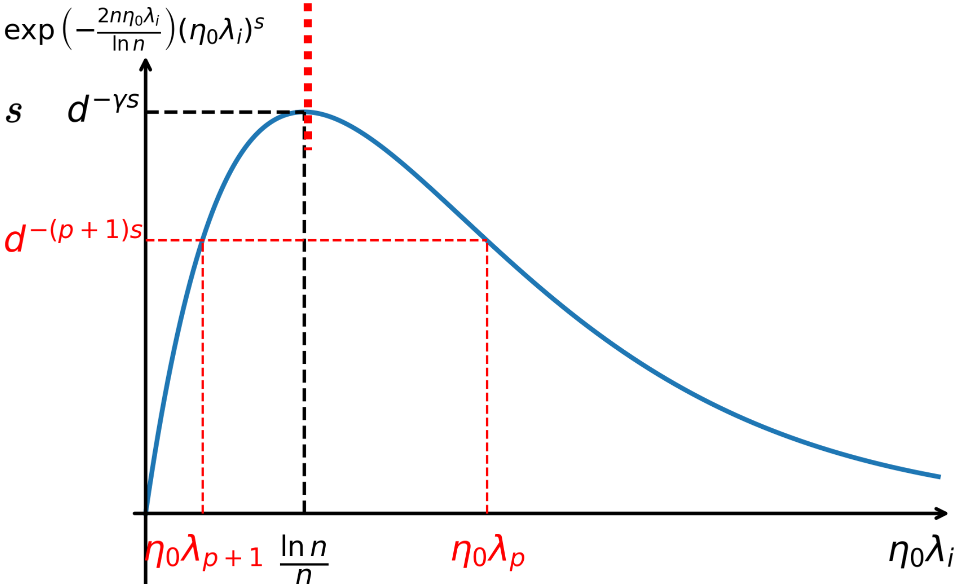
$$\lambda_k \asymp d^{-k}, \quad N(d,k) \asymp d^{-k}, \quad p = \left\lceil \frac{\gamma}{s+1} - 1 \right\rceil, \eta_0 \asymp d^{-\gamma+p} \ln n \ln d.$$

Variance： $\lesssim (1 + d + \cdots + d^p) \cdot \frac{1}{n} \asymp d^{-\gamma+p}$ $\quad + \quad$ $\lesssim \lambda_{p+1} n \eta_0^2 \asymp d^{-\gamma+p-1}$

$\lambda_0 \sim 1$	$d \times \lambda_1 \sim d^{-1}$	…………	$d^p \times \lambda_p \sim d^{-p}$	$d^{p+1} \times \lambda_{p+1} \sim d^{-(p+1)}$	…………

Bias： $\lesssim \eta_0^{-s} \max_{i \geq 0} \exp(-2\frac{n\eta_0\lambda_i}{\ln n})(\eta_0\lambda_i)^s$

$\lesssim d^{-(p+1)s}$



Excess Risk： $d^{-((\gamma-p)\wedge(p+1)s)}$

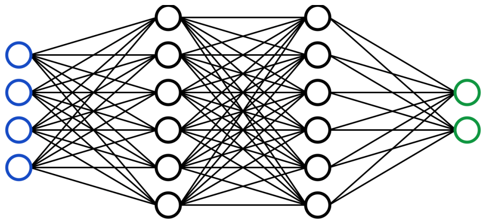
北京大學
PEKING UNIVERSITY

Optimal Algorithms in Linear Regression under Covariate Shift: On the Importance of Precondition

Out-of-Distribution Generalization

Source Data

Training on $\mathbb{P}^S$



Model

Test on $\mathbb{P}^T$

Excess Risk

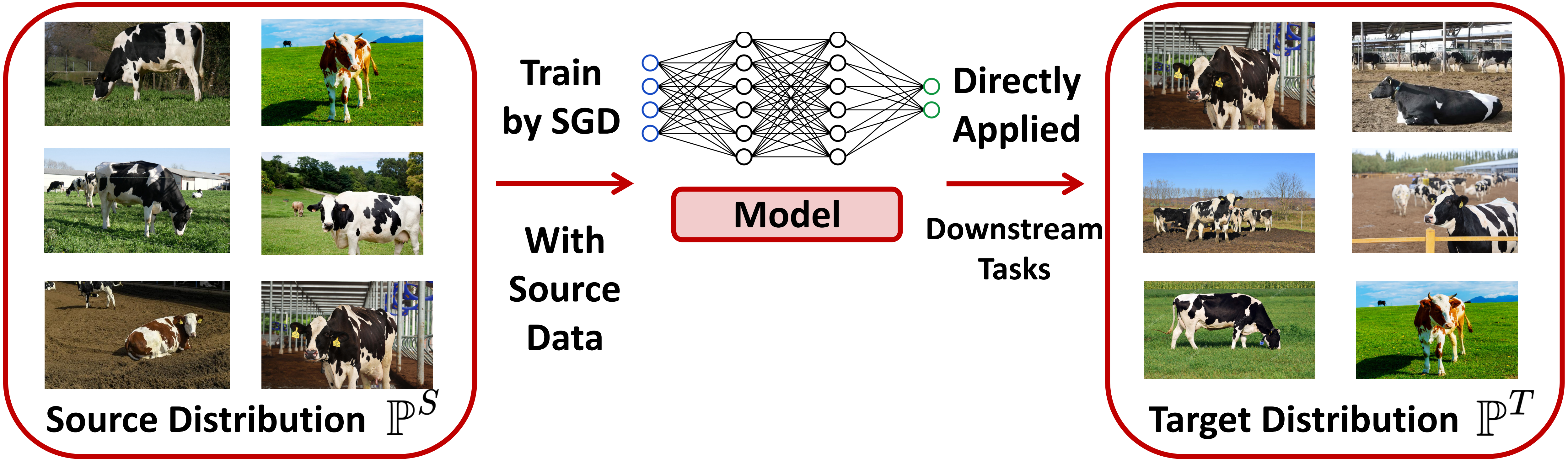
- **Common Assumption:** $\mathbb{P}^S = \mathbb{P}^T$
- **Out-of-Distribution:** $\mathbb{P}^S \neq \mathbb{P}^T$

Source Distribution $\mathbb{P}^S$

Target Distribution $\mathbb{P}^T$

Covariate Shift

➢ **Covariate Shift:** $\mathbb{P}^S \neq \mathbb{P}^T, \quad \text{but} \quad \mathbb{P}^S_{y|\mathbf{x}} = \mathbb{P}^T_{y|\mathbf{x}}.$



Framework

What are the Out-of-Distribution generalization capabilities and limitations of models trained with SGD?

- What is the min-max optimal algorithm under covariate shift?
- What is the excess risk of ASGD under covariate shift?
- When can ASGD achieve optimality under covariate shift?
- A unified view of ASGD as a prediction estimator.

Covariate Shift under High-Dimensional Linear Regression

Source Data: $\{(\mathbf{x}_i, y_i)\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} \mathbb{P}^S_{\mathbf{x}\times y}$ and $\mathbf{x} \in \mathbb{R}^d, d$ may $\gg n$.

Source & Target Covariance Matrix: $\mathbf{S} = \mathbb{E}_{\mathbb{P}^S_{\mathbf{x}}}[\mathbf{x}\mathbf{x}^\top], \quad \mathbf{T} = \mathbb{E}_{\mathbb{P}^T_{\mathbf{x}}}[\mathbf{x}\mathbf{x}^\top].$

Regression Model:

$$\begin{cases} \arg\min_{\mathbf{w}\in\mathbb{R}^d} \mathcal{E}_{\mathcal{S}} = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \mathcal{E}_{\mathcal{T}} = \mathbf{w}^*. \\ \epsilon = y - \mathbf{x}^\top \mathbf{w}^*, \quad \mathbb{E}\left[\epsilon^2 \mathbf{x}\mathbf{x}^\top\right] \preceq \sigma^2 \mathbf{S}. \\ W = \left\{\mathbf{w}^* \in \mathbb{R}^d : \|\mathbf{w}^*\|^2_{\mathbf{M}} \le 1\right\}. \end{cases}$$

Target Generalization Error: $\mathcal{E}_{\mathcal{T}}(\mathbf{w}) = \frac{1}{2}\mathbb{E}_{\mathbb{P}^T_{\mathbf{x}\times y}}(y - \langle \mathbf{w}, \mathbf{x}\rangle)^2.$

Target Excess Risk: $\mathcal{R}_{\mathcal{T}}(\mathbf{w}) = \frac{1}{2}\left(\mathcal{E}_{\mathcal{T}}(\mathbf{w}) - \min_{\mathbf{w}} \mathcal{E}_{\mathcal{T}}(\mathbf{w})\right) = \frac{1}{2}\|\mathbf{w} - \mathbf{w}^*\|^2_{\mathbf{T}}.$

Optimal Algorithm

What is the min-max *optimal* algorithm under covariate shift?

The Power of Precondition

Consider $W = \left\{ \mathbf{w}^* \in \mathbb{R}^d : \|\mathbf{w}^*\|_2^2 \leq 1 \right\}.$

OLS Estimator: $\hat{\mathbf{w}} = \frac{1}{n}\mathbf{S}^{-1}\sum_{i=1}^{n}\mathbf{x}_i y_i$ **Variance:** $\frac{1}{n}\mathbf{S}^{-1}$ **Bias:** $\approx \mathbf{0}$

Precondition by A: $\hat{\mathbf{w}}_{\mathbf{A}} = \frac{1}{n}\mathbf{A}\mathbf{S}^{-1}\sum_{i=1}^{n}\mathbf{x}_i y_i$ **Variance:** $\frac{\sigma^2}{n}\mathbf{A}\mathbf{S}^{-1}\mathbf{A}^\top$ **Bias:** $(\mathbf{I}-\mathbf{A})\mathbf{w}^*(\mathbf{w}^*)^\top(\mathbf{I}-\mathbf{A})^\top$

Target Excess Risk of $\hat{\mathbf{w}}_{\mathbf{A}}$:

$$\mathbb{E}_{\tilde{P}^{\otimes n}}||\hat{\mathbf{w}}_{\mathbf{A}} - \mathbf{w}^*||_{\mathbf{T}}^2 \lesssim \langle \mathbf{T}, (\mathbf{I}-\mathbf{A})\mathbf{w}^*(\mathbf{w}^*)^\top(\mathbf{I}-\mathbf{A})^\top \rangle + \frac{\sigma^2}{n}\langle \mathbf{T}, \mathbf{A}\mathbf{S}^{-1}\mathbf{A}^\top \rangle$$

Optimal Algorithm

What is the min-max *optimal* algorithm under covariate shift?

Optimal Preconditioner Design

$$\min_{\mathbf{A}\in\mathbb{R}^{d\times d}} \max_{\|\mathbf{w}^*\|_{\mathbf{M}}^2\le 1} \left\langle \mathbf{T}, (\mathbf{I}-\mathbf{A})\mathbf{w}^*(\mathbf{w}^*)^\top(\mathbf{I}-\mathbf{A})^\top \right\rangle + \frac{\sigma^2}{n}\left\langle \mathbf{T}, \mathbf{A}\mathbf{S}^{-1}\mathbf{A}^\top \right\rangle$$
$$= \min_{\mathbf{A}\in\mathbb{R}^{d\times d}} ||(\mathbf{I}-\mathbf{A})^\top\mathbf{T}(\mathbf{I}-\mathbf{A})|| + \frac{\sigma^2}{n}\left\langle \mathbf{T}, \mathbf{A}(\mathbf{S})^{-1}\mathbf{A}^\top \right\rangle$$

General Optimal Preconditioner

$$\mathbf{A} = \operatorname*{argmin}_{\mathbf{A}\in\mathbb{R}^{d\times d}} ||(\mathbf{I}-\mathbf{A})^\top\mathbf{T}'(\mathbf{I}-\mathbf{A})|| + \frac{\sigma^2}{n}\left\langle \mathbf{T}', \mathbf{A}(\mathbf{S}')^{-1}\mathbf{A}^\top \right\rangle,$$

$$\mathbf{S}' = \mathbf{M}^{-1/2}\,\mathbf{S}\,\mathbf{M}^{-1/2}, \quad \mathbf{T}' = \mathbf{M}^{-1/2}\,\mathbf{T}\,\mathbf{M}^{-1/2}, \quad W = \left\{\mathbf{w}^* \in \mathbb{R}^d : \|\mathbf{w}^*\|_{\mathbf{M}}^2 \le 1\right\}.$$

- This is actually a min-max estimator.
- Pathak et al (2024) already achieved the results.

ASGD Excess Risk

$$\mathbf{w}_n^{\text{sgd}} \approx \text{ a specialized preconditioner } \hat{\mathbf{w}}_{\mathbf{A}} \text{ by } \mathbf{A} = \begin{bmatrix} \mathbf{I}_{k^*} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{bmatrix}!$$

ASGD Target Excess Risk Bound:

$$\mathbb{E}_{\tilde{P}^{\otimes n}} ||\mathbf{w}_n^{\text{sgd}} - \mathbf{w}^*||_{\mathbf{T}}^2$$

$$\lesssim \frac{\sigma^2}{n} \left\langle \mathbf{T}', \begin{bmatrix} \mathbf{I}_{k^*} & \mathbf{O} \\ \mathbf{O} & n(\gamma+\delta)\mathrm{diag}\{\lambda_i\}_{i=k^*+1}^{d} \end{bmatrix} \left(\mathbf{S}'\right)^{-1} \begin{bmatrix} \mathbf{I}_{k^*} & \mathbf{O} \\ \mathbf{O} & n(\gamma+\delta)\mathrm{diag}\{\lambda_i\}_{i=k^*+1}^{d} \end{bmatrix} \right\rangle$$

$$+ \left|\left| \left(\mathbf{I} - \begin{bmatrix} \mathbf{I}_{k^*} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{bmatrix}\right) \mathbf{T}' \left(\mathbf{I} - \begin{bmatrix} \mathbf{I}_{k^*} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{bmatrix}\right) \right|\right| + \frac{1}{n^2} \left|\left| \begin{bmatrix} \mathbf{I}_{k^*} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{bmatrix} \mathbf{T}' \begin{bmatrix} \mathbf{I}_{k^*} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{bmatrix} \right|\right|.$$

- $k^* = \max\left\{k : \lambda_k > \frac{32\ln^2 n}{(\gamma+\delta)n}\right\}$ refers to the effective dimension.
- $\mathbf{T}' = \mathbf{M}^{-1/2}\,\mathbf{T}\,\mathbf{M}^{-1/2}, \quad \mathbf{S}' = \mathbf{M}^{-1/2}\,\mathbf{S}\,\mathbf{M}^{-1/2}, \quad \mathbf{S} = \mathrm{diag}\{\lambda_i\}_{i=1}^{d}.$

Optimality Under the Diagonal Dominant Condition

r-smooth Class $\mathcal{Q}$

There exists a constant $C > 0$, such that for any $\mathbb{P}_{\mathbf{x}}^{T} \in \mathcal{Q}$, $\mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{\mathbf{x}}^{T}}\left[\mathbf{x}\mathbf{x}^{\top}\right] \preceq C\mathbf{S}^{r+1}$.

There exist a constant $c > 0$ and $\mathbb{P}_{\mathbf{x}}^{T_0} \in \mathcal{Q}$, such that $\mathbf{T}_0 \succeq c\mathbf{S}^{r+1}$.

Examples of r-smooth Class

Density Ratio Bounded Class: Let $B > 0$ be a constant and $\mathcal{Q} = \left\{Q : \mathrm{d}\mathbb{P}_{\mathbf{x}}^{T}/\mathrm{d}\mathbb{P}_{\mathbf{x}}^{S} \leq B\right\}$. (0-smooth class)

Gaussian D_{KL} Bounded Class: Let $\epsilon > 0$ and $\mathcal{Q} = \left\{\mathbb{P}_{\mathbf{x}}^{T} : D_{\mathrm{KL}}(\mathbb{P}_{\mathbf{x}}^{T}||\mathbb{P}_{\mathbf{x}}^{S}) < \epsilon,\ \mathbb{P}_{\mathbf{x}}^{T} \text{ is Gaussian}\right\}$. (0-smooth class)

Power-law Anisotropic Covariance Structures

(1) $\lambda_i \eqsim i^{-a}$ with $a > 1$. (2) $\mathbf{M} = \mathrm{diag}\{m_i\}_{i=1}^{d},\ \ m_i \eqsim \lambda_i^{1-s}$.

ASGD Optimal Region

(1) $s \geq 1 - \dfrac{1}{a}$, vanilla SGD optimal.

(2) Broader $1 - \dfrac{1}{a} > s > \dfrac{(a-1)^2}{a(2a-1)}$, ASGD achieves optimality.

Optimality Beyond the Diagonal Dominant Condition

Asymptotic Settings

$$\text{When } \frac{n}{\ln^2 n} \geq \frac{1}{\lambda_d}, ||\mathbf{w}_*||_2^2 \leq 1, \text{ SGD achieve optimal rate } \mathcal{O}\left(\frac{\mathbf{S}^{-1}\mathbf{T}}{n}\right).$$

Rank-1 Target Covariance

$$(1) \quad \lambda_i \eqsim i^{-a}, \quad \mathbf{T} = \mathbf{w}\mathbf{w}^\top \text{ where } \mathbf{w} \in \mathbb{R}^d \text{ and } \mathbf{w}_i \eqsim i^{-\gamma}.$$

$$(2) \quad W = \left\{\mathbf{w}^* \in \mathbb{R}^d : \|\mathbf{w}^*\|^2_{\mathbf{S}^{1-\frac{\alpha+2\beta-1}{\alpha}}} \leq 1\right\}.$$

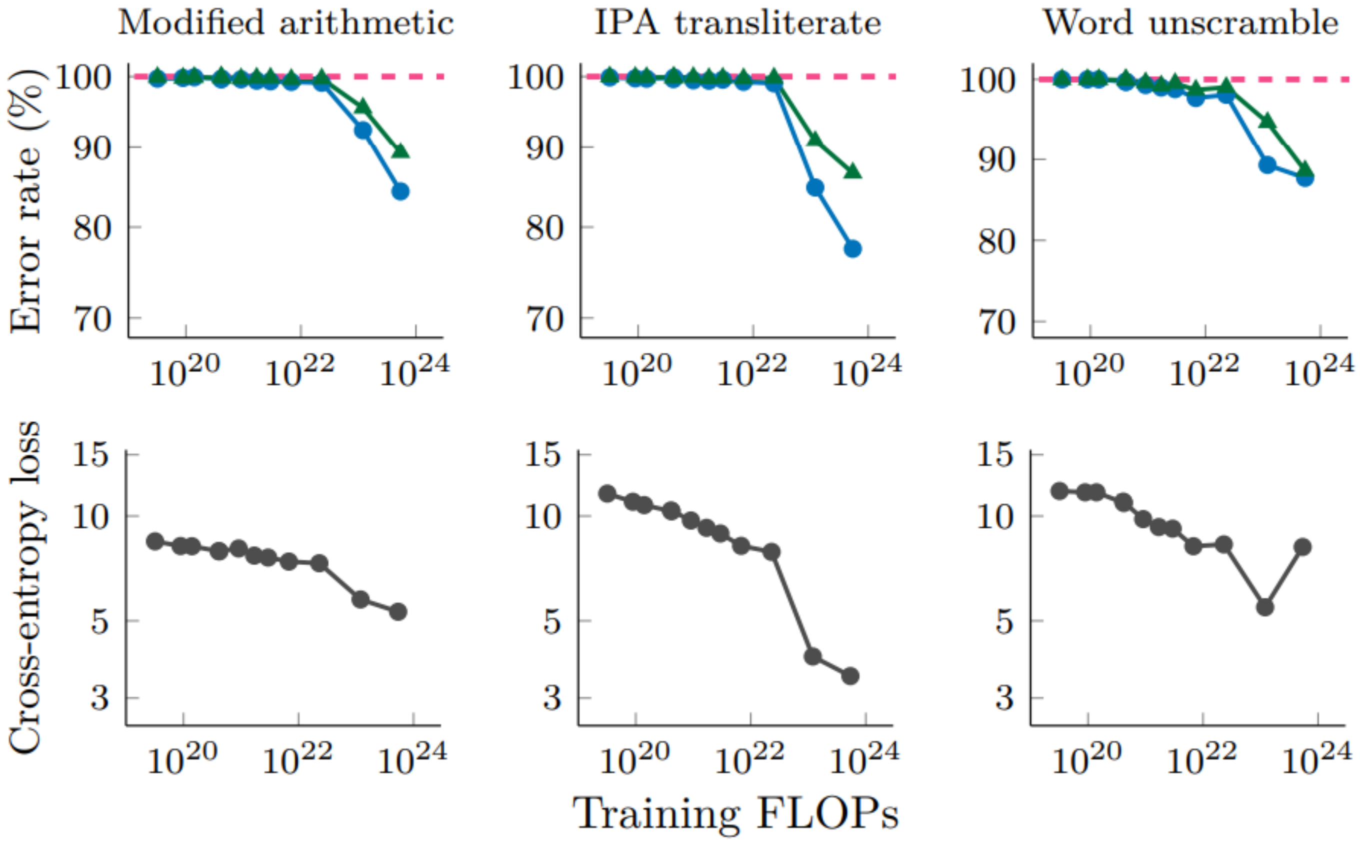
Optimal Rate: $\frac{1}{n}, \quad \alpha + 1 \leq 2\gamma; \quad n^{-\frac{2(\beta+\gamma-1)}{\alpha+2\beta-1}}, \quad 2\beta \geq \alpha + 1.$

ASGD Optimal Region: $2\beta \geq \alpha + 1, \quad \text{vanilla SGD is optimal;}$

$$\text{broader } 1 + \frac{\alpha^2}{2\alpha - 1} < 2\beta < \alpha + 1, \text{ ASGD is optimal.}$$

What is Emergence?

Emergence: A ***sharp and unpredictable decrease*** in test loss with respect to **model size, sample size, training FLOPs**, or **specific task types**...



Wei et al. (2022)

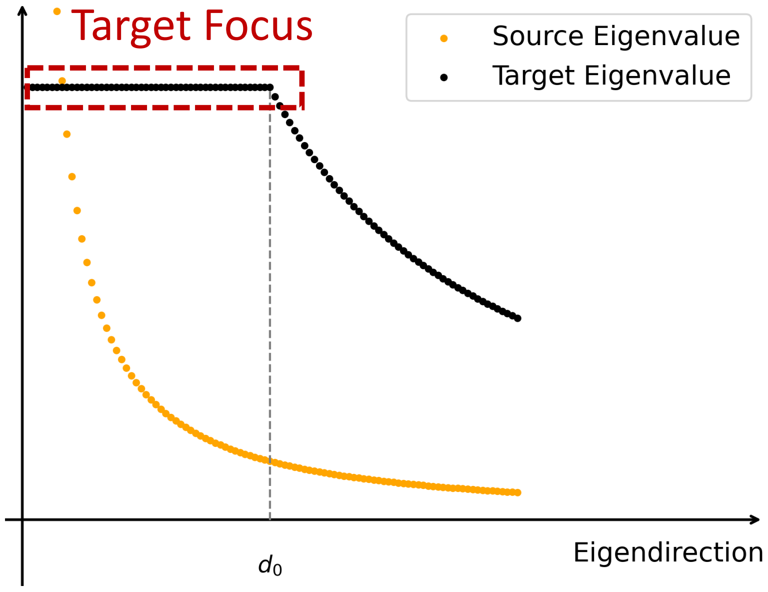
Emergence: When and Why?

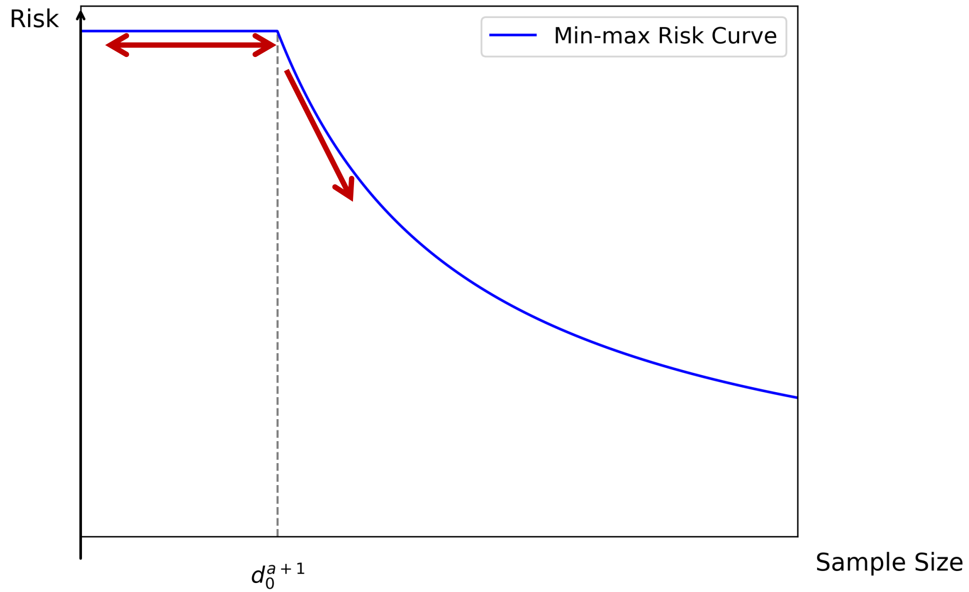
Are Emergent Abilities of Large Language Models a Mirage?

- The target distribution demands high-accuracy estimation in localized regions.
- Even the min-max risk can exhibit a sharp decrease once the sample size n exceeds a certain threshold.

Example: $\mathbf{S} = \mathrm{diag}\{i^{-a}\}_{i=1}^{d}$, $\mathbf{T} = \mathrm{diag}\left\{\{1\}_{i=1}^{d_0}, \{i^{-a}\}_{i=d_0+1}^{d}\right\}$, $\|\mathbf{w}^*\|_2^2 \leq 1.$

Min-max Rate: $\mathcal{O}(1),\ n \lesssim d_0^{a+1}$; $\mathcal{O}\left(n^{-\frac{a}{1+a}}\right), n \gtrsim d_0^{a+1}.$



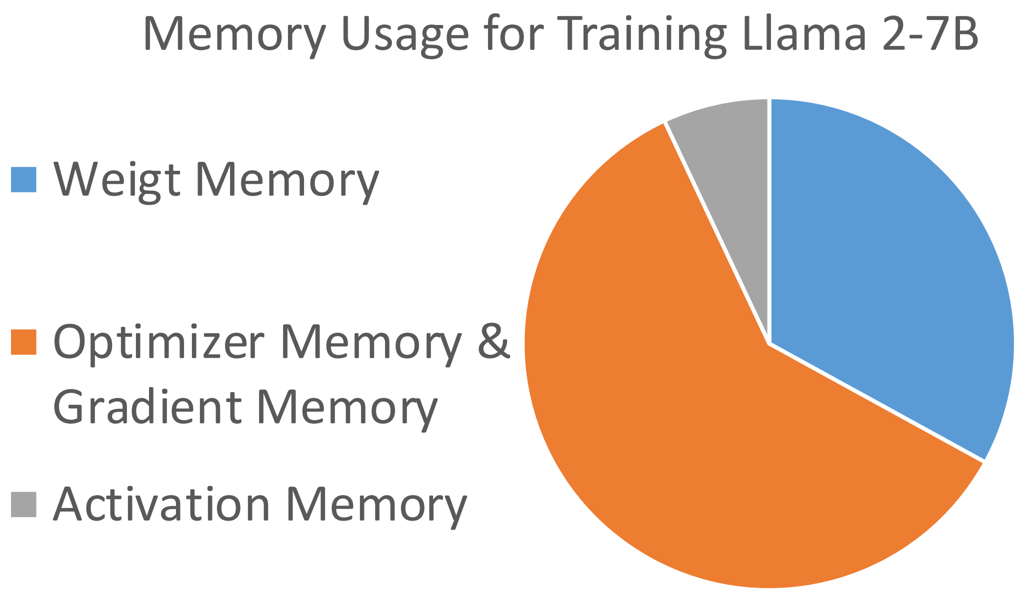


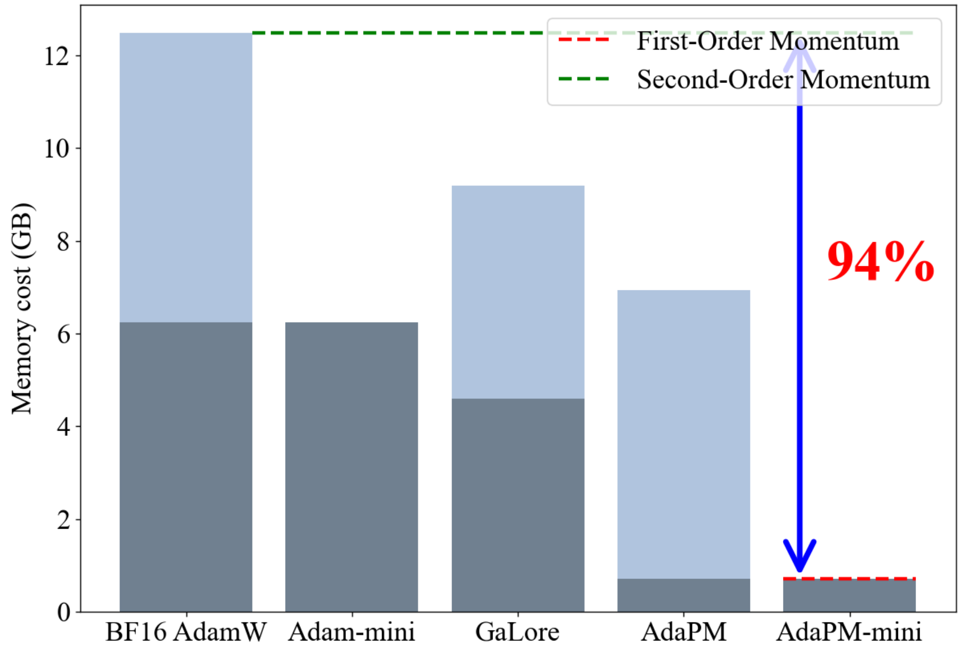
北京大學
PEKING UNIVERSITY

AdaPM: a Partial Momentum Algorithm for LLM Training

Introduction

- Adam requires the memory for its optimizer states:





- Design effective optimizers that require less memory.
  - It requires fewer GPUs to train a model with a desired size, leading to substantial savings in both cost and energy.
  - It can ease the burden of CPU offloading and model sharding, which in turn, can enhance the throughput and accelerate the training process.

Momentum Can be Redundant

A Motivating Example:

- **Regressing** $y = \langle W, x \rangle + \text{noise}$ **with covariate** $x \sim N(0, \Sigma)$

  Σ is diagonal with $\Sigma_{ii} = i^{-a}$ $\Sigma_{ii} W_i^2 = i^{-b}$

- **Accelerated SGD with momentum $1 - \beta$,**

  $\beta = 1$ → **vanilla SGD**

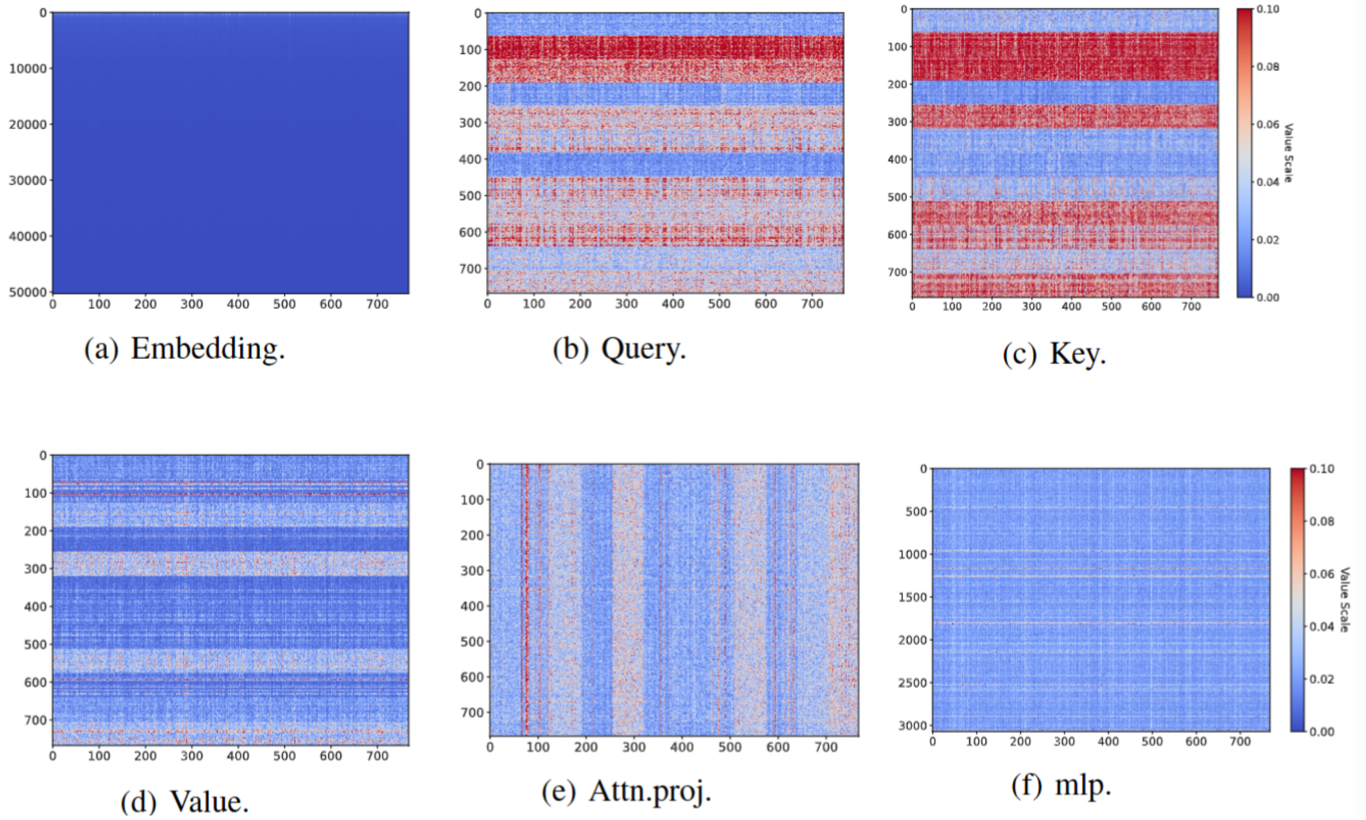
- **Then the excess risk is given by**

$$\tilde{O}\left(\underbrace{T^{\frac{1}{a}-1}\, \beta^{\frac{1}{a^2}-\frac{1}{a}}}_{\text{Algorithm's Variance}} + \underbrace{T^{\frac{1}{a}-\frac{b}{a}}\, \beta^{\left(\frac{1}{a^2}-\frac{1}{a}\right)(1-b)}}_{\text{Algorithm's bias}} \right)$$

- **Momentum will enlarge Algorithm's variance!**
- **Large variance and small bias: momentum is redundant**

Algorithm

- **Sparse Gradients:** most of the gradient matrices in Embedding and Attn.proj are filled by near-zero values.



(a) Embedding. (b) Query. (c) Key.
(d) Value. (e) Attn.proj. (f) mlp.

- Partition approach of Transformers:
- **Embedding and Attention Output Projection Blocks:** disable momentum.
- **Query, Key, and MLP blocks:** debiased low-rank approximation.
- **Value layers:** full momentum.

Algorithm

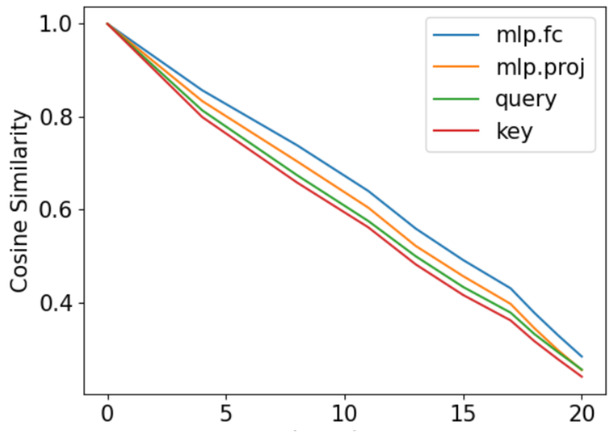
- Debiased Low-rank Estimator
  - Low-rank approximation of the momentum $\mathrm{m_t}$

$$\mathbf{m}_t \leftarrow (1-\beta_1)\nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t) + \beta_1 \mathbf{L}_{t-1}\mathbf{R}_{t-1}$$

$$\mathbf{L}_t\mathbf{R}_t \in \arg\min_{\mathbf{L},\mathbf{R}} \left\| \mathbf{LR} - \left((1-\beta_1)\tilde{\nabla} f(W_t) + \beta_1 \mathbf{L}_{t-1}\mathbf{R}_{t-1} \right) \right\|^2 .$$

  - One-step residual

$$r_t = \mathbf{L}_t\mathbf{R}_t - \left((1-\beta_1)\tilde{\nabla} f(\mathbf{W}_t) + \beta_1 \mathbf{L}_t\mathbf{R}_t \right)$$



  - Refine the momentum estimate

Assumption 1 (Stationary Residuals). *The one-step residuals $\{r_t\}_{t\geq 1}$ are identically distributed across iterations, i.e., $r_t \overset{d}{=} r_{t'}$ for all $t, t' \geq 1$.*

$$\mathbf{m}_t^c = \mathbf{m}_t - \frac{r_t}{1-\beta_1}$$

Algorithm

Require: Weight-decay coefficient λ, decay rates of momentum β_1, β_2, rank of the momentum approximation matrices r and learning rate schedule $\{\eta_t\}_{t=1}^T$

Obtain mini-batch gradient $\nabla f(\mathbf{W}_t, \xi_t)$

$\mathbf{m}_t \leftarrow (1 - \beta_1)\nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t) + \beta_1 \mathbf{L}_{t-1}\mathbf{R}_{t-1}$

- Standard second-order momentum update

$$\mathbf{v}_t = \beta_2 \mathbf{v}_{t-1} + (1 - \beta_2)[\nabla f(\mathbf{W}_t, \boldsymbol{\xi}_t)]^{\odot 2}$$

- Approximation residual

$$\mathbf{L}_t, \mathbf{R}_t = \arg\min_{\mathbf{L},\mathbf{R}} \|\mathbf{L}\mathbf{R} - \mathbf{m}_t\|_F^2$$

$$r_t = \mathbf{m}_t - \mathbf{L}_t\mathbf{R}_t$$

- Bias correction for low-rank momentum

$$\mathbf{m}_t^c = \mathbf{m}_t - \frac{r_t}{1-\beta_1}$$

$$\mathbf{W}_{t+1} = \mathbf{W}_t - \eta_t \left(\mathrm{clip}\left(\frac{\mathbf{m}_t^c}{\sqrt{\mathbf{v}_t + \epsilon}}, 1\right) + \lambda \mathbf{x}_t\right)$$

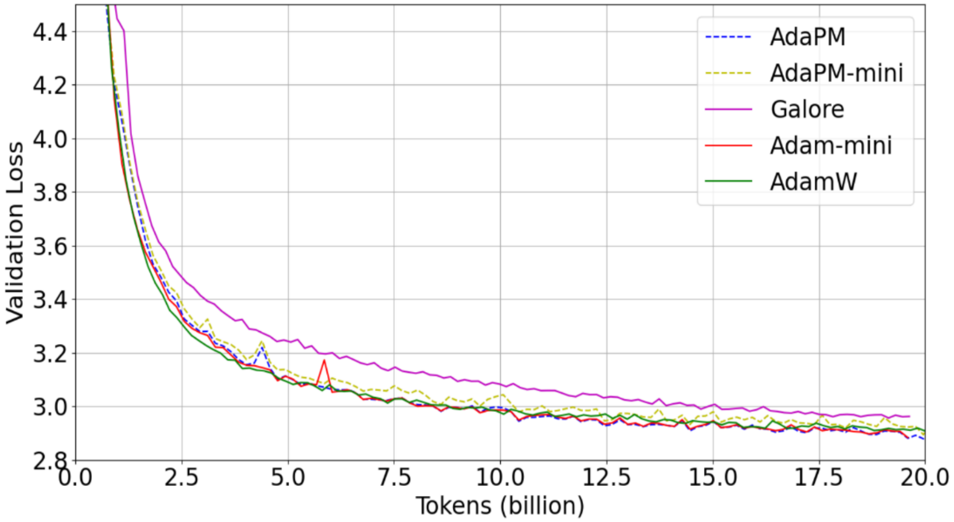
Experiment

1、Pretraining on GPT-2

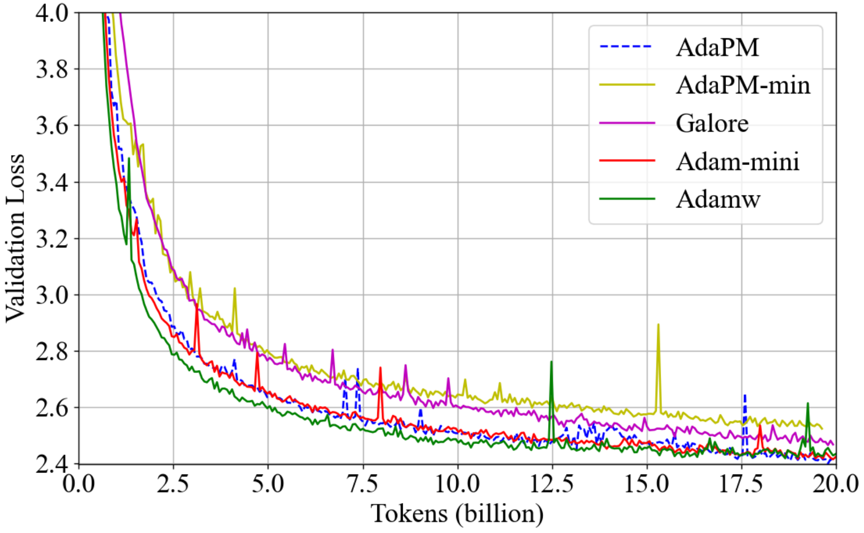
- Model: GPT-2 124M, 1.5B
- Dataset: Openwebtext(17G)

Algorithm	GPT-2-1.5B Optimizer Memory	GPT-2-1.5B GPU Hours
Adam	12.48G	26.67
Adam-mini	6.24G	20.32
AdaPM	6.98G (↓ **44%**)	22.11 (↓ **17%**)
AdaPM-mini	0.74G (↓ **94%**)	17.92 (↓ **33%**)

The loss curves of Adapm closely resemble those of AdamW while reduces momentum memory consumption to approximately 44% of baseline requirements.



GPT-2 124M



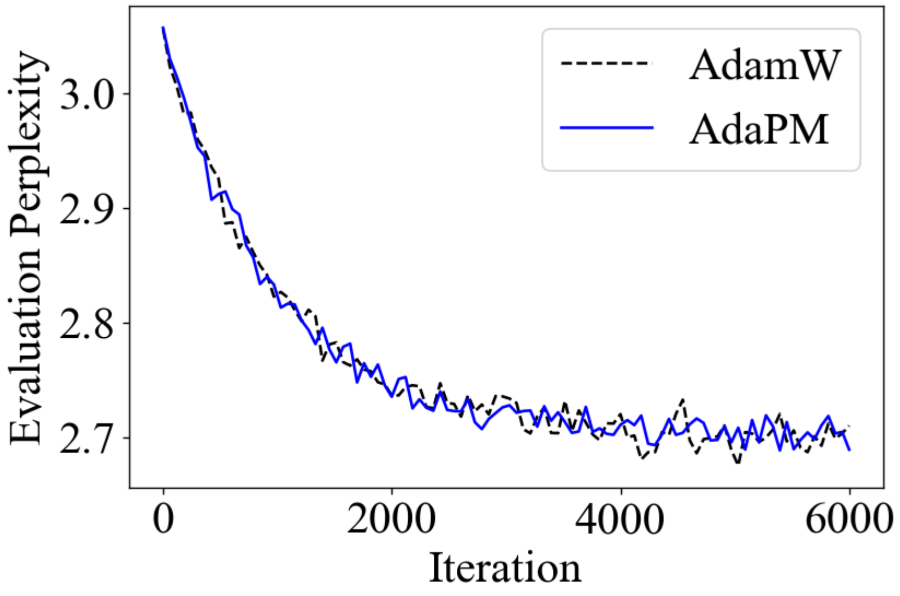
GPT-2 1.5B

Experiment

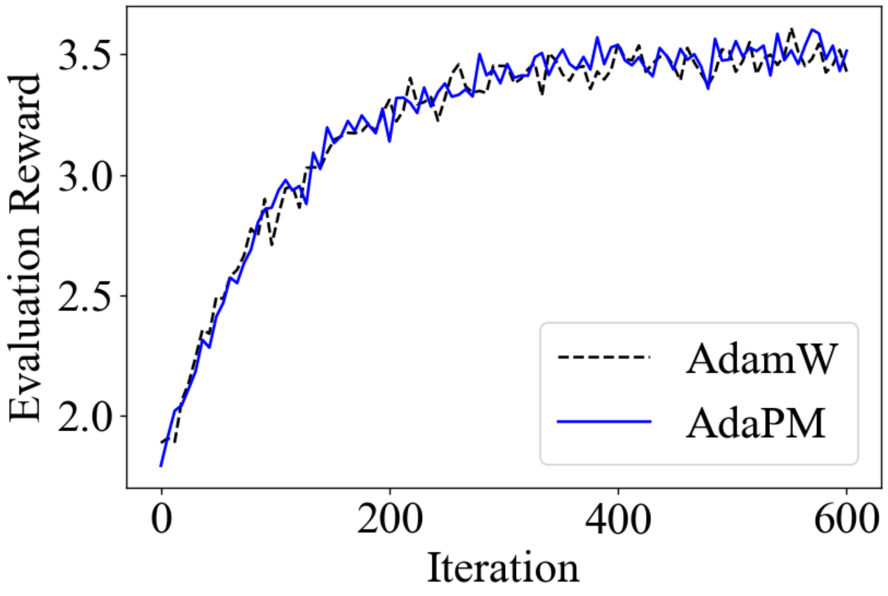
2、 Finetuning

- Base Model: llama-3-8b
- Dataset: Ultrafeedback

AdaPM performs on par or better than AdamW.



SFT



RLHF

北京大學
PEKING UNIVERSITY

Thanks