Classifying high-dimensional Gaussian mixtures: Where kernel methods fail and neural networks succeed

Jaehyoung Hong

ICML 2021 Poster

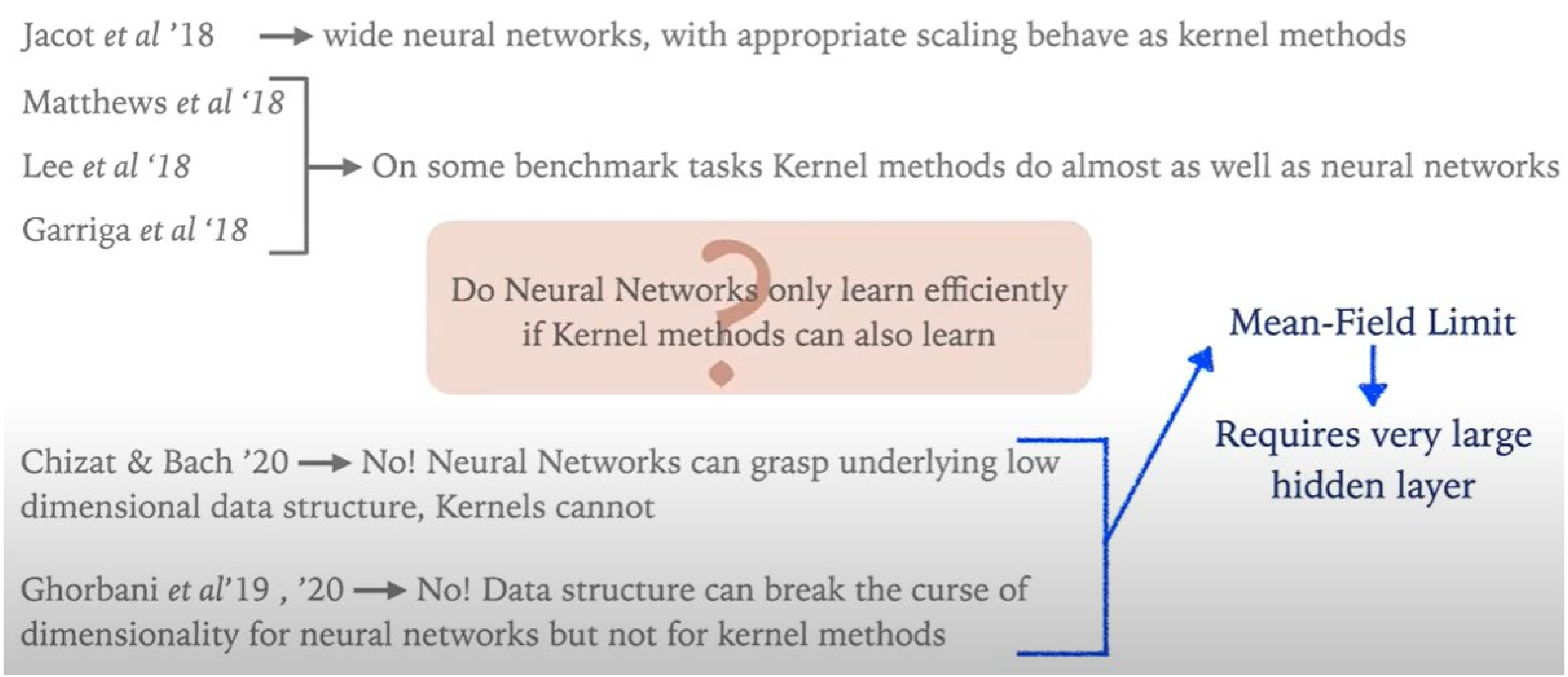
Kernel methods shows competitive performance to Neural network for some task

Jacot *et al* ’18 → wide neural networks, with appropriate scaling behave as kernel methods

Matthews *et al* ‘18
Lee *et al* ‘18
Garriga *et al* ‘18
→ On some benchmark tasks Kernel methods do almost as well as neural networks

Do Neural Networks only learn efficiently if Kernel methods can also learn

Kernel methods did not beat the Neural network for high-dimensional setting



✓ Today's topic: NN > Kernel method for $O(1)$ hidden nodes, 2 layers

Compare two-layer neural network(2LNN) and random features

- $\#(sample) = N, \#(input\ dimension) = D, \#(random\ features) = P$
- $Input\ x = (x_r) \in \mathbb{R}^D, Labels\ y \in \{-1,1\}$: Binary classification (for simplicity)

$\boldsymbol{\epsilon_c(\theta)}$
$= \mathbb{E}\mathbf{H}[-\boldsymbol{y\phi_\theta(x)}]$

$N = O(D^2)$ needed for RF while $N = O(D)$ for 2LNN

1. 2LNN ($\phi_\theta^L: \mathbb{R}^D \to \mathbb{R}$)

$$x \Rightarrow \lambda^k \equiv \frac{1}{\sqrt{D}}\sum_{r=1}^{D} w_r^k x_r \Rightarrow \phi_\theta^L(x) = \sum_{k=1}^{K} v^k g(\lambda^k)$$

- ✓ Trainable weight $w = (w_r^k) \in \mathbb{R}^{K\times D}, v = (v^k) \in \mathbb{R}^K$
- ✓ $\frac{K}{D} = O(1)$: Small node setting, $t \equiv \frac{N}{D} = O(1)$: High dimensional setting ($D \to \infty$)

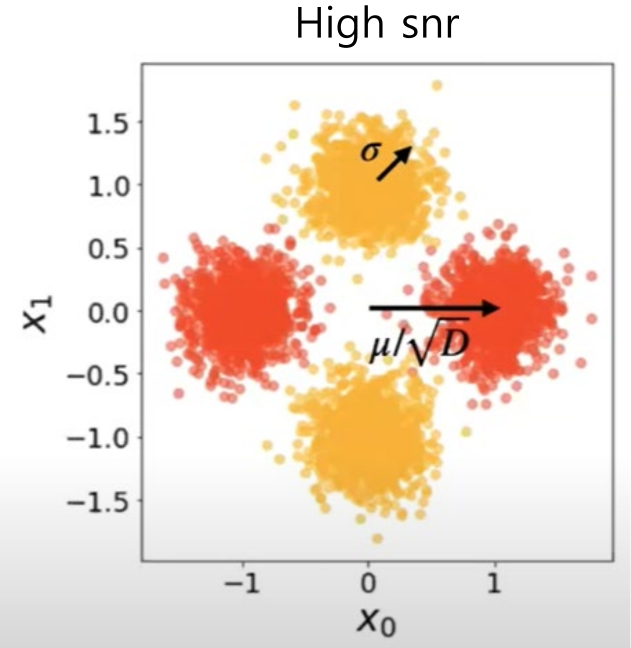
2. Random Features ($\phi_\theta^R: \mathbb{R}^D \to \mathbb{R}$)

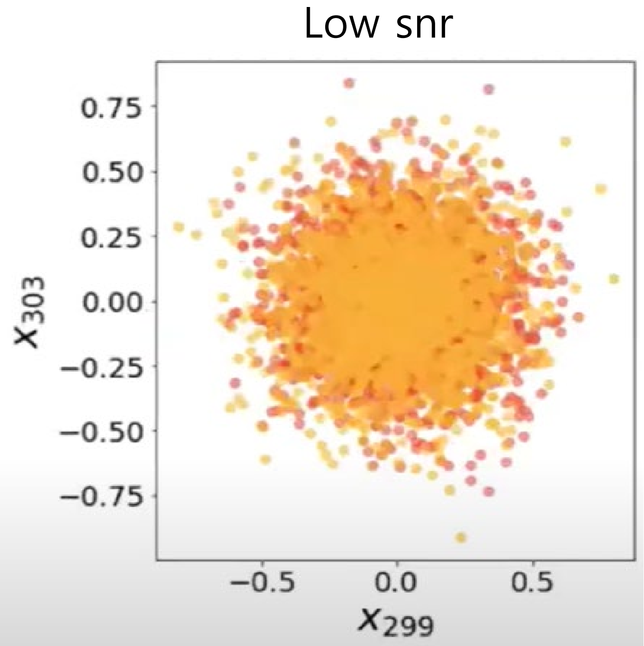
$$x \Rightarrow u_i \equiv \frac{1}{\sqrt{D}}\sum_{r=1}^{D} F_{ir} x_r \Rightarrow z_i = \psi(u_i) \Rightarrow \phi_\theta^R(z) = \frac{1}{\sqrt{P}}\sum_{i=1}^{P} w_i z_i$$

- ✓ Trainable weight $w = (w_i) \in \mathbb{R}^P$
- ✓ $\gamma \equiv \frac{P}{D} = O(1)$, $N > P$

RF is similar to neural network for High signal to noise ratio(SNR) while worse for Low SNR

- **XOR inputs**



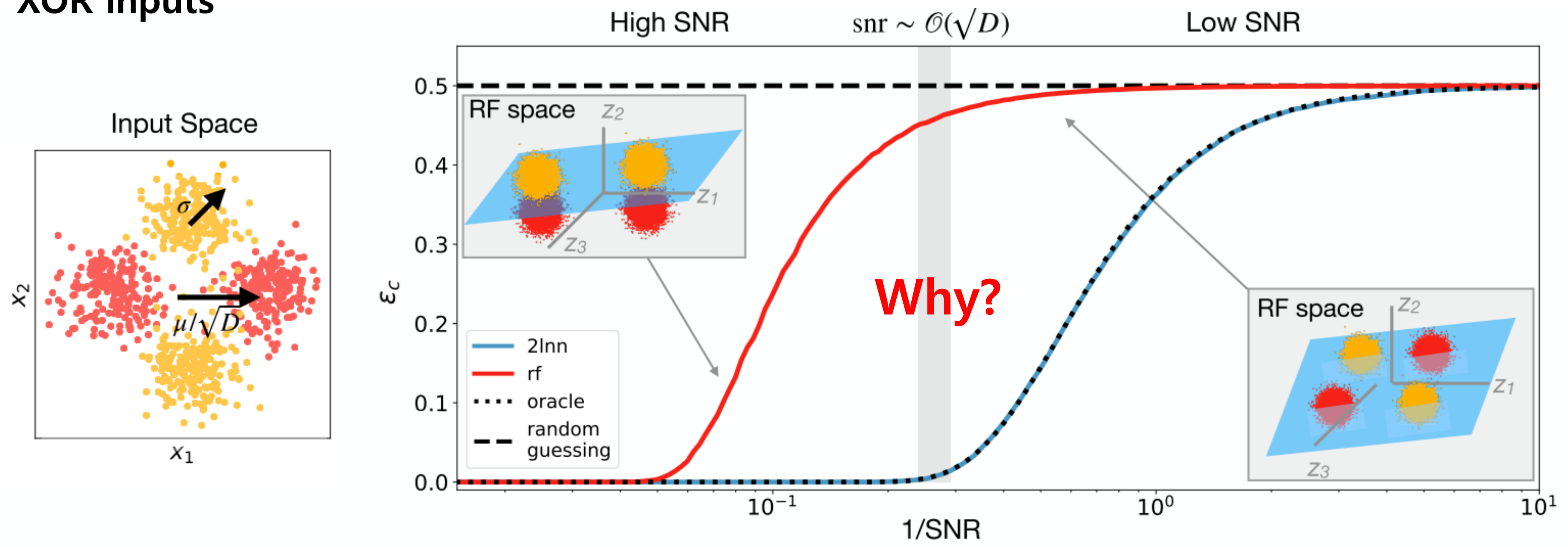


✓ $snr = |\mu|/\sqrt{D}\sigma$ for input $x_i = \mu_i + \sigma z_i$ ($z_i \sim N(0,1)$), XOR: $\mu_+ = \left(\pm\sqrt{D}, 0, \ldots, 0\right), \mu_- = (0, \pm\sqrt{D}, 0, \ldots, 0)$

✓ $oracle$: Known about μ_i and labels x_i to that of nearest μ_i

RF is similar to neural network for High signal to noise ration(SNR) while worse for Low SNR

- **XOR inputs**



✓ $snr = |\mu|/\sqrt{D}\sigma$ for input $x_i = \mu_i + \sigma z_i$ $(z_i \sim N(0,1))$, XOR: $\mu_+ = (\pm\sqrt{D}, 0, \dots, 0), \mu_- = (0, \pm\sqrt{D}, 0, \dots, 0)$

✓ $oracle$: Known about μ_i and labels to nearest μ_i

Setup for dynamics of 2LNN: Moments of λ^k

- **2LNN for GM classification**

  ✓ Sample (x, y) is drawn from $q(x, y) = q(y)q(x|y)$,
  $q(x, y) = \sum_{\alpha \in S(y)} \mathcal{P}_\alpha N_\alpha(x)\,, N_\alpha(x) \sim N(\frac{\mu^\alpha}{\sqrt{D}}, \Omega^\alpha)$

  ✓ Online learning, $\Delta = \sum_{k=1}^{K} v^k g(\lambda^k) - y$

$$\mathrm{d}w_i^k = \underbrace{-\frac{\eta}{\sqrt{D}} v^k \Delta g'(\lambda^k) x_i}_{\text{SGD}} \underbrace{- \frac{\eta}{\sqrt{D}} \kappa w_i^k}_{L^2\text{-Regularization}},$$
$$\mathrm{d}v^k = \underbrace{-\frac{\eta}{D} g(\lambda^k) \Delta}_{\text{SGD}} \underbrace{- \frac{\eta}{D} \kappa v^k}_{L^2\text{-Regularization}},$$

L^2-Regularization

SGD

  ✓ Static

$$\text{pmse}(\theta) = \mathop{\mathbb{E}}_{q(x,y)} \left(y - \phi_\theta(x)\right)^2$$
$$= \sum_y \sum_{\alpha \in \mathcal{S}(y_i)} q(y_i) \mathcal{P}_\alpha \mathop{\mathbb{E}}_{\alpha} \left[\sum_k v^k g(\lambda^k) - y\right]^2$$

Setup for dynamics of 2LNN: Moments of λ^k

- **2LNN for GM classification**

Average over the inputs x

$$\mathrm{pmse}(\theta) = \mathop{\mathbb{E}}_{x}\left(\sum_{k=1}^{K} v^k g\left(\frac{w^k x}{\sqrt{D}}\right) - y\right)^2 \xrightarrow{\text{Zoom into each cluster}} \mathrm{pmse}(\theta) = \sum_{clusters} \underbrace{\mathop{\mathbb{E}}_{x}\left[\left(\sum_{k=1}^{K} v^k g\left(\frac{w^k x}{\sqrt{D}}\right) - y\right)^2 | x \sim \mathcal{N}\left(\frac{\mu_\alpha}{\sqrt{D}}, \Sigma_\alpha\right)\right]}_{\text{Expectation over cluster } \alpha}$$

$$\mathrm{pmse}(\theta) = \sum_{clusters} \mathop{\mathbb{E}}_{\lambda_\alpha}\left[\left(\sum_{k=1}^{K} v^k g\left(\lambda^k\right) - y\right)^2 | x \sim \mathcal{N}\left(\frac{\mu_\alpha}{\sqrt{D}}, \Sigma_\alpha\right)\right]$$

Average over the local fields

✓ λ^k are jointly Gaussian when averages are evaluated over just a single distribution in the mixture

✓ $pmse$ is determined by the moments of λ^k

Setup for dynamics of 2LNN: Moments of λ^k

- **2LNN for GM classification**

✓ Order parameters: Moments of λ^k

$$M_\alpha^k \equiv \mathop{\mathbb{E}}_\alpha \lambda^k = \frac{1}{D}\sum_r w_r^k \mu_r^\alpha,$$

$$Q_\alpha^{k\ell} \equiv \mathop{\mathsf{Cov}}_\alpha \left(\lambda^k, \lambda^\ell\right) = \frac{1}{D}\sum_{r,s} w_r^k \Omega_{rs}^\alpha w_s^\ell.$$

✓ Since any average over a Gaussian is fcn of two moments,

$$\lim_{D\to\infty} \mathsf{pmse}(\theta) \to \mathsf{pmse}(Q, M, v).$$

✓ Pmse of network ≈ (Q, M, v) dynamics

Dynamics of weight are determined by order parameters

- Moments of functions of weakly correlated variables has explicit form
- $t = N/D$ can be regarded as continuous time variable for online learning

Let r.v. $x, y \in \mathbb{R}$, jointly Gaussian & weakly correlated:

$$P(x,y) = \frac{1}{2\pi\sqrt{\det M_2}} \exp\left[-\frac{1}{2}\begin{pmatrix} x-\bar{x} & y-\bar{y}\end{pmatrix} M_2^{-1}\begin{pmatrix} x-\bar{x} \\ y-\bar{y}\end{pmatrix}\right], \qquad M_2 = \begin{pmatrix} C_x & \epsilon M_{12} \\ \epsilon M_{12} & C_y \end{pmatrix}$$

$$= \frac{1}{2\pi\sqrt{C_x C_y}} \mathrm{e}^{-\frac{1}{2C_x}(x-\bar{x})^2 - \frac{1}{2C_y}(y-\bar{y})^2}\left[1 - \epsilon(x-\bar{x})\left(C_x^{-1}M_{12}C_y^{-1}\right)(y-\bar{y}) + O\left(\epsilon^2\right)\right]$$

$$\underset{(x,y)}{\mathbb{E}}[f(x)g(y)] = \underset{x}{\mathbb{E}}[f(x)]\,\underset{y}{\mathbb{E}}[g(y)] + \epsilon\,\underset{x}{\mathbb{E}}[f(x)(x-\bar{x})]\left(C_x^{-1}M_{12}C_y^{-1}\right)\underset{y}{\mathbb{E}}[g(y)(y-\bar{y})] + O(\epsilon^2).$$

- Dynamics of second layer ($\Omega_\alpha = \Omega$)

$$\mathrm{d}v^k = -\frac{\eta}{D} g(\lambda^k)\Delta - \frac{\eta}{D}\kappa v^k,$$

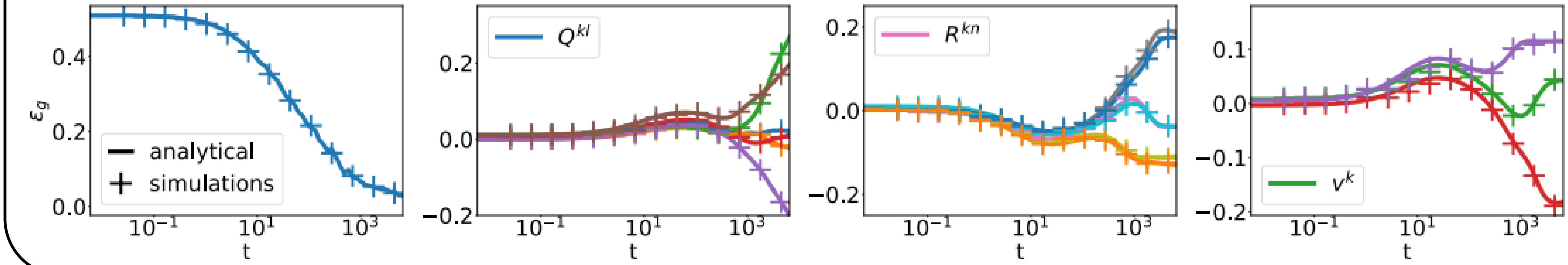
$$\mathbb{E}\,\mathrm{d}v^k = \sum_{\alpha\in\mathcal{S}(+)} \mathcal{P}_\alpha\,\mathrm{d}v^k_{\alpha+} + \sum_{\alpha\in\mathcal{S}(-)} \mathcal{P}_\alpha\,\mathrm{d}v^k_{\alpha-},$$

$$\mathrm{d}v^k_\alpha = \frac{\eta}{D}\underset{\alpha}{\mathbb{E}}\, y_\alpha g(\lambda^k) - \frac{\eta}{D}\sum_j v^j\,\underset{\alpha}{\mathbb{E}}\, g(\lambda^k)g(\lambda^j) - \frac{\eta}{D}\kappa v^k.$$

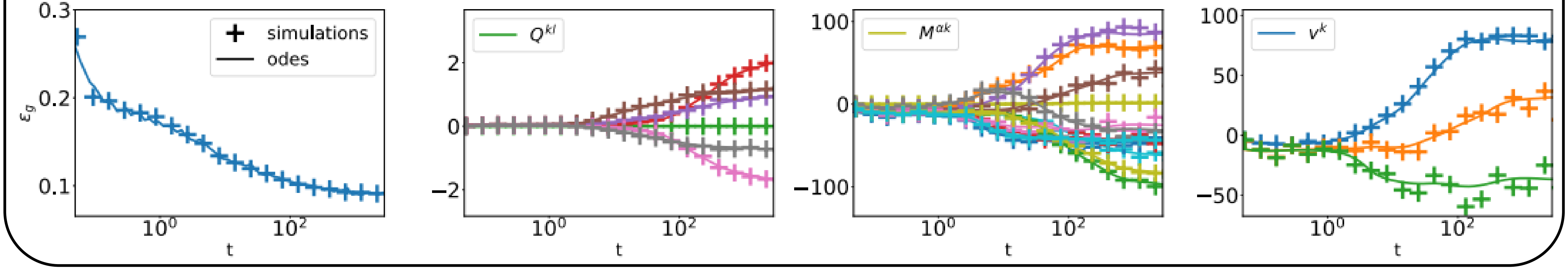
✓ Dynamics of weight are determined by order parameters /
Expectation of activation function is estimated by MC methods

Order parameters and weights from ODE are agreed with simulation(Single run of SGD)

- Simulation data(GM mixture)

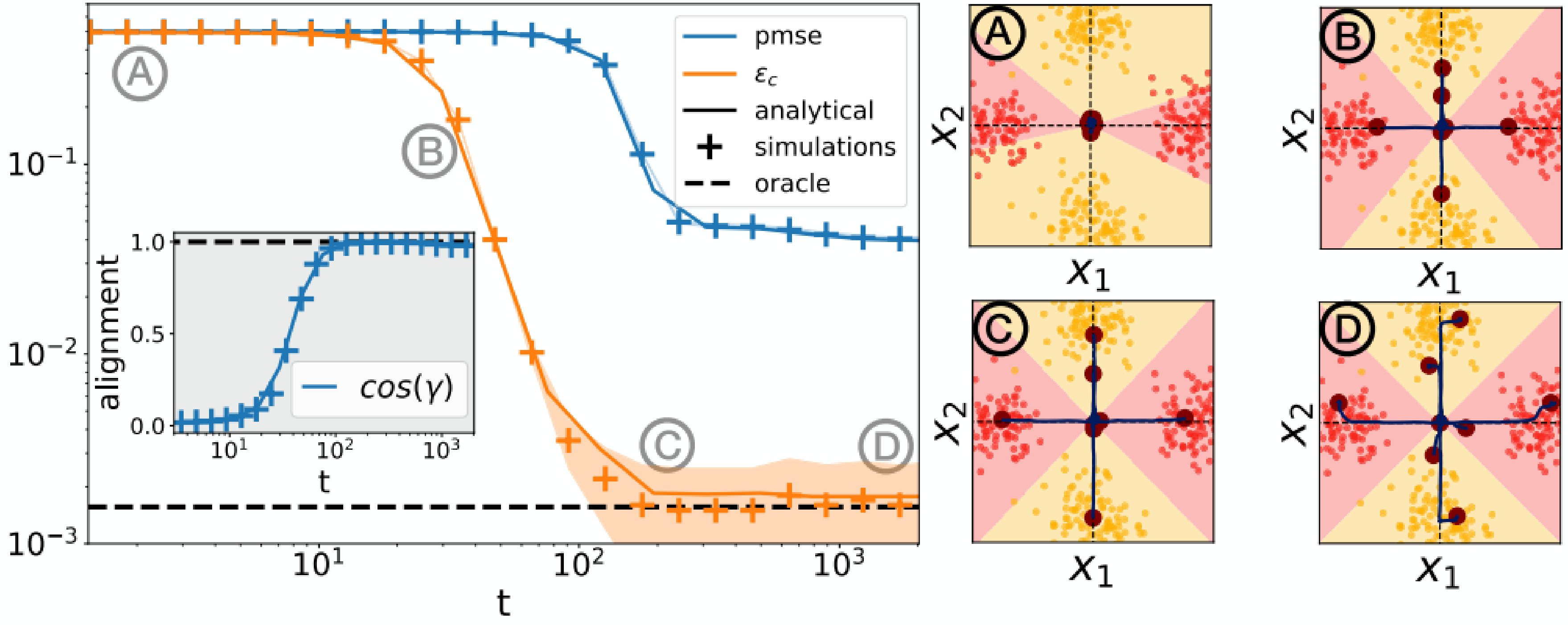


- FashionMNIST, Ω_α is different for α



2LNN learns XOR inputs when first weight vectors approach the four means

- (Left) Error plot (Right): First weight projected onto the sample space (Blood dots)

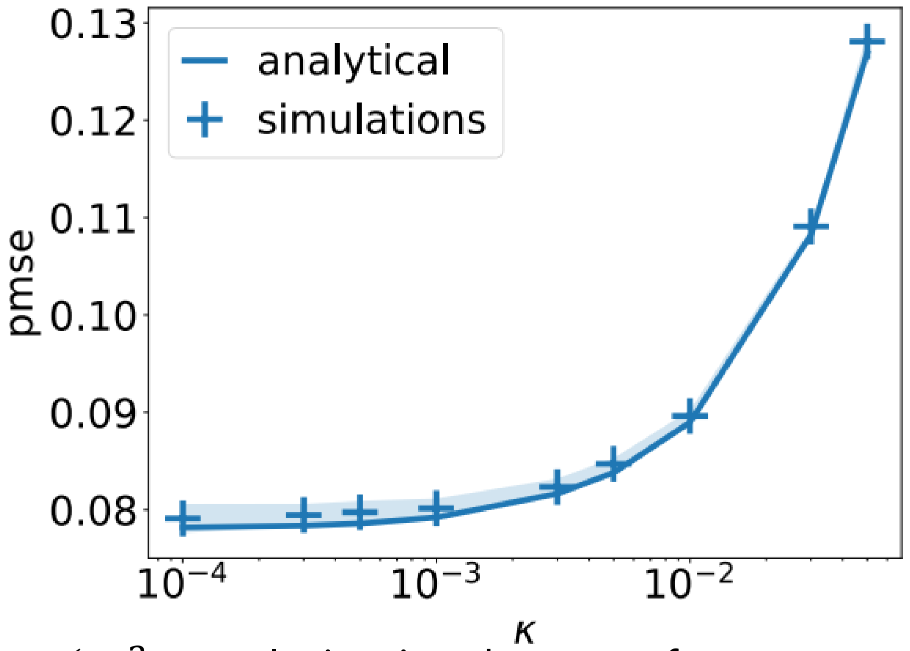


C: First weight finds four means (Maximal angle)

D: Decrease pmse

Predicting long-time performance using ansatz also agreed with simulation

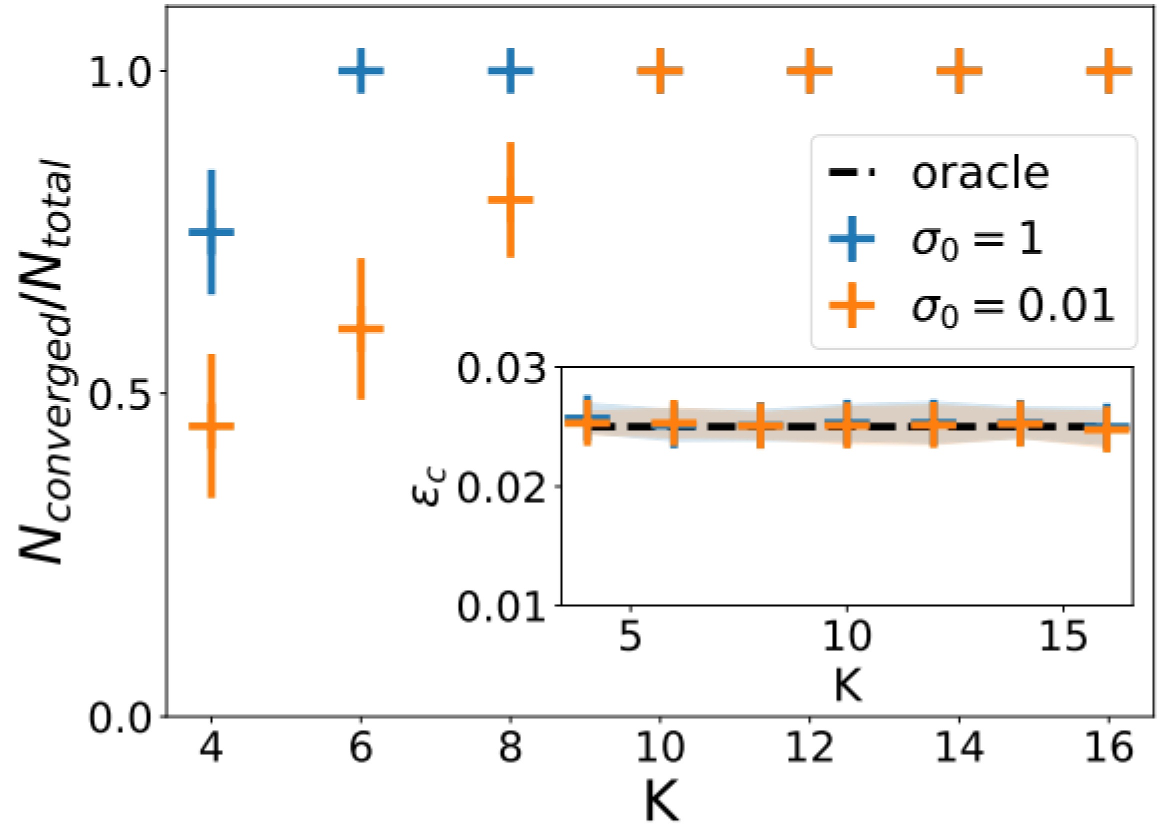
- Long time performance of ODE: Asymptotic fixed point
- Scale $K^2 + 4K$ equations for each time step
- Ansatz (e.g. symmetry) reduce parameters to K



✓ L^2-regularization hurt performance

✓ $K = 4$ is enough

Over-parametrization do not improve test error, but has acceleration effect



Dynamics of RF

- Assume $N \gg P$, for any finite D,P, running the algorithm upto convergence (≈Taking the limit $t \to \infty$)

- ρ_τ are the eigenvalues of the feature's covariance matrix $\Omega_{ij} = \mathbb{E} z_i z_j$, with associated eigenvector Γ_τ. $\widetilde{\Phi}_\tau \equiv \sum_{i=1}^{P} \Gamma_{\tau i} \Phi_i / \sqrt{P}$

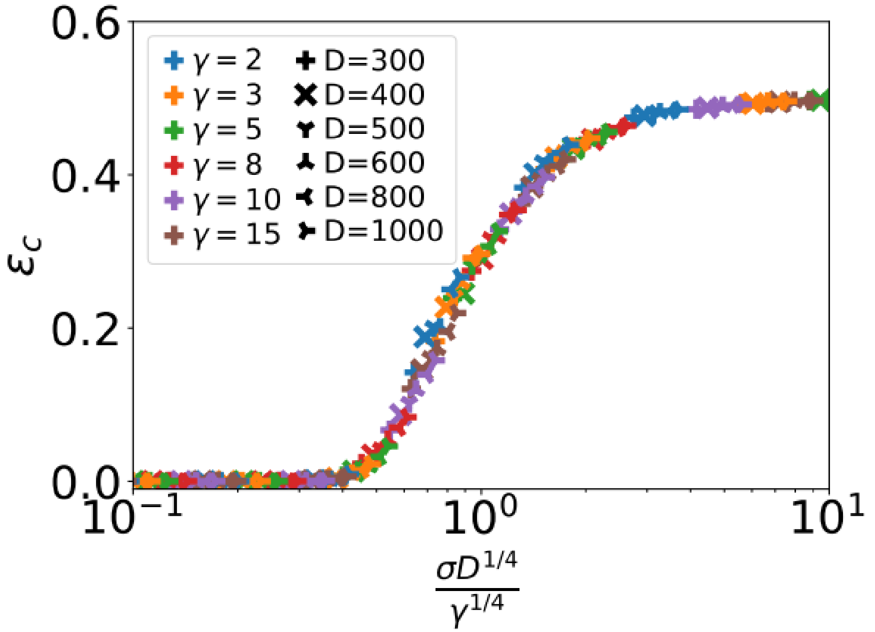
$$\mathrm{pmse}_{t\to\infty} = \frac{1}{2}\left(1 - \sum_\tau \frac{\tilde{\Phi}_\tau^2}{\rho_\tau}\right),$$

$$M_\alpha = \sum_{i=1}^{P} \frac{\hat{w}_i \,\mathbb{E}_{\alpha}[z_i]}{\sqrt{P}}, \quad Q_\alpha = \sum_{i=1}^{P} \frac{\hat{w}_i \hat{w}_j}{P} \underset{\alpha}{\mathrm{Cov}}(z, z).$$

$$\epsilon_{ct\to\infty} = \frac{1}{2}\left(1 - \sum_\alpha \mathcal{P}_\alpha \, y \,\mathrm{erf}\left(\frac{M_\alpha}{\sqrt{2Q_\alpha}}\right)\right)$$

Classification error of RF for various values shows that $P \approx D^2$ features are required

- Classification error when $N \gg P$



✓ Graph shows for good performance $P = O(D^2)$ while $N \gg P$ & $N = O(D)$ makes it impossible

✓ Indeed, classification error is fcn of $\sigma D^{\frac{1}{2}} / \min(N, P)^{1/4}$

✓ If $N = O(D)$ & $\sigma \gg N^{\frac{1}{4}} / D^{1/2}$,
performance degrades to no more than random guess

In Low SNR setting($snr \sim O(1)$), the transformation of the means is only linear

- $\frac{|\mu|}{\sqrt{D}} \sim O(1)$ & $\sigma \sim O(1) \rightarrow \frac{F_{ir}\mu_r}{D} \sim O(\frac{1}{\sqrt{D}})$

$$a \equiv \mathbb{E}\, \psi(\sigma\zeta), \quad b \equiv \mathbb{E}\, \zeta\psi(\sigma\zeta), \quad c^2 \equiv \mathbb{E}\, \psi(\sigma\zeta)^2$$

$$\mathbb{E}\, z_i = a + b \sum_{r=1}^{D} \frac{F_{ir}\mu_r}{\sigma D}$$

$$\mathrm{cov}(z_i, z_j) = \begin{cases} c^2 - a^2, & i = j\,, \\ b^2 \sum_r \frac{F_{ir}F_{jr}}{D} & i \neq j. \end{cases}$$

✓ Transformation of the means is only linear

✓ RF cannot separate XOR inputs with Low SNR setting in feature space

✓ RF only separate when the centres of data are separated enough

Convergence of RF to kernel methods shows that kernel methods also fails for Low SNR setting

- Rahimi &Recht, 2008;2009 ($D, P \to \infty$ and then $\gamma \to \infty$)

$$K(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{P} \sum_{i=1}^{P} \mathbb{E}_{F} \left[\psi \left(\sum_{r=1}^{D} \frac{x_r F_{ir}}{\sqrt{D}} \right) \psi \left(\sum_{s=1}^{D} \frac{y_s F_{is}}{\sqrt{D}} \right) \right]$$

- With Low SNR setting

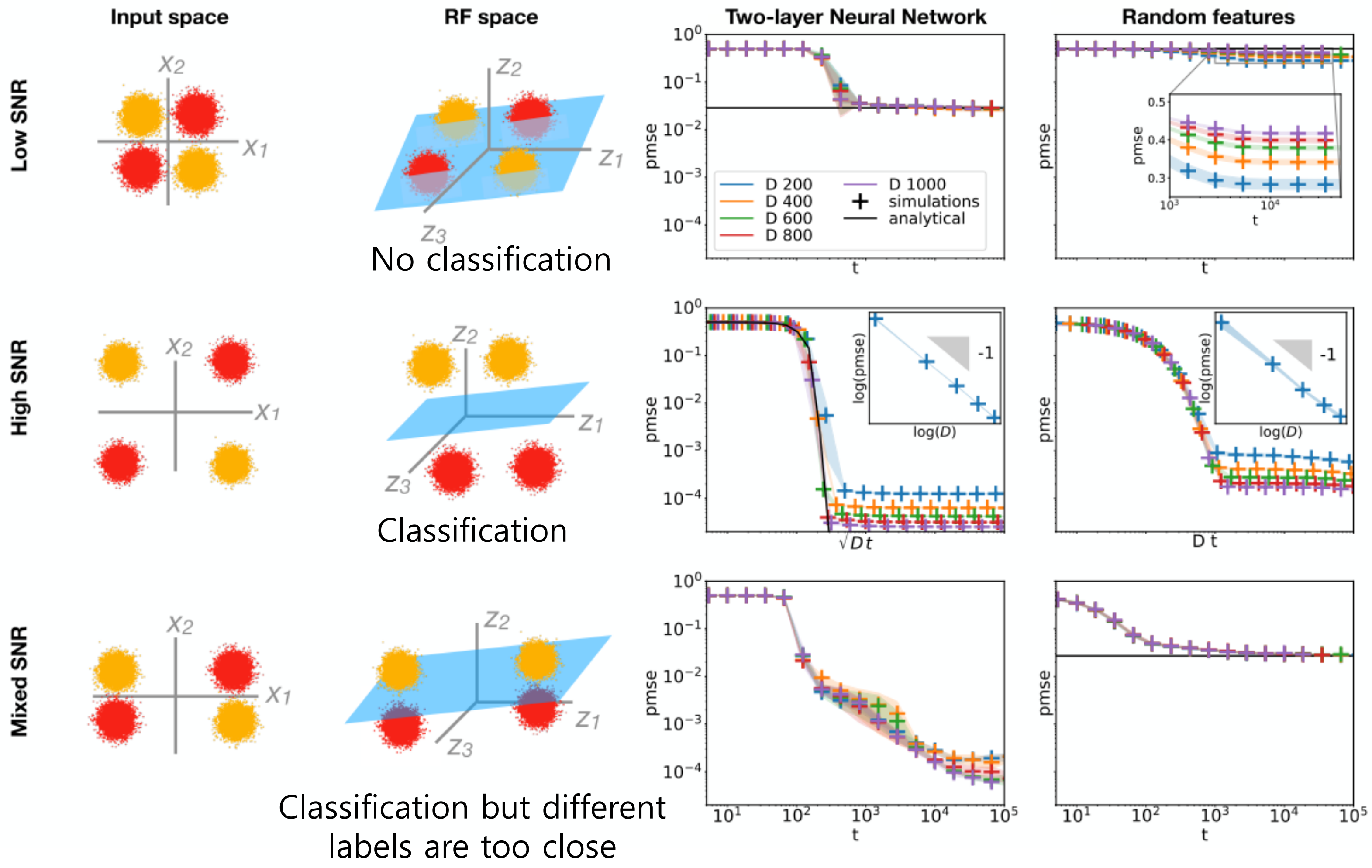
$$c^2 = \mathbb{E}\, K(\sigma \boldsymbol{\omega}_1, \sigma \boldsymbol{\omega}_1), \quad a^2 = \mathbb{E}\, K(\sigma \boldsymbol{\omega}_1, \sigma \boldsymbol{\omega}_2),$$

$$b^2 = D\sigma^2 \left[-a^2 + \mathbb{E}\, K \left(\frac{\boldsymbol{\mu}}{\sqrt{D}} + \sigma \boldsymbol{\omega}_1, \frac{\boldsymbol{\mu}}{\sqrt{D}} + \sigma \boldsymbol{\omega}_2 \right) \right]$$

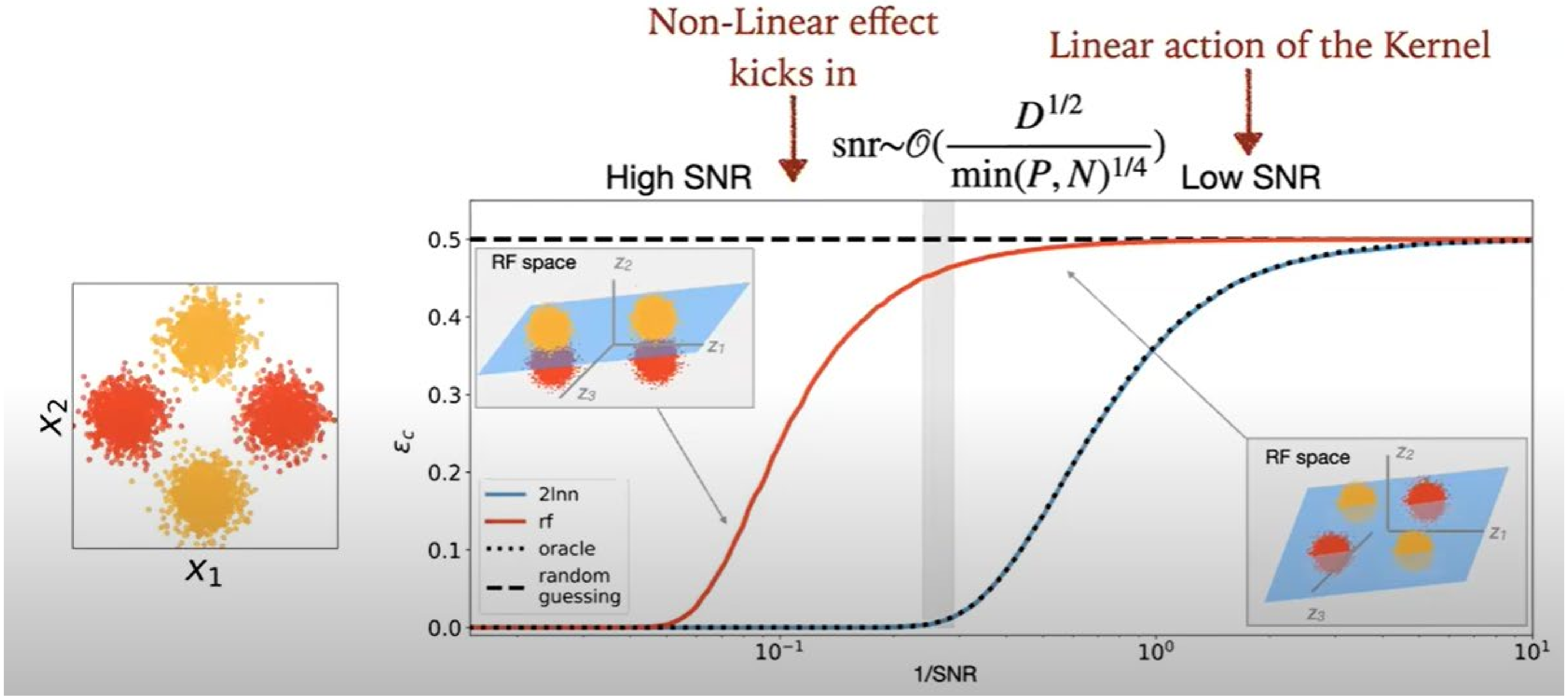
$$\mathbb{E}\, z_i = a + b \sum_{r=1}^{D} \frac{F_{ir} \mu_r}{\sigma D}$$

$$\mathrm{cov}(z_i, z_j) = \begin{cases} c^2 - a^2, & i = j\,, \\ b^2 \sum_r \frac{F_{ir} F_{jr}}{D} & i \neq j. \end{cases}$$

2LNN is better than RF for GM classification



Summary



Non-Linear effect kicks in
Linear action of the Kernel
$\mathrm{snr} \sim \mathcal{O}\left(\frac{D^{1/2}}{\min(P,N)^{1/4}}\right)$
High SNR
Low SNR
RF space
z_1
z_2
z_3
2lnn
rf
oracle
random guessing
ε_c
1/SNR
x_1
x_2
0.0
0.1
0.2
0.3
0.4
0.5
10^{-1}
10^{0}
10^{1}