# The role of over-parameterization in machine learning: the good, the bad, the ugly

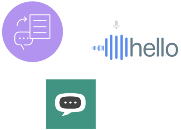
- from a function space view

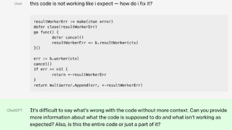
**Fanghui Liu**

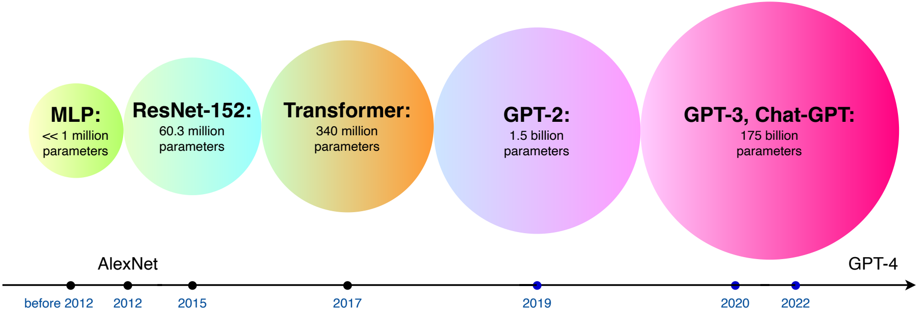
Department of Computer Science, University of Warwick, UK
Centre for Discrete Mathematics and its Applications (DIMAP), Warwick

at AAAI New Faculty Highlights 2024, Vancouver

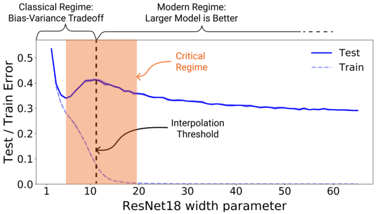
# Over-parameterization: more parameters than training data



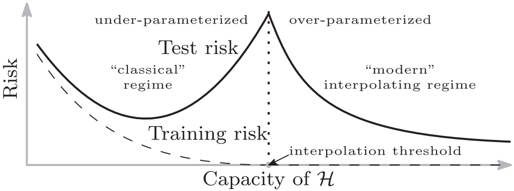



# Surprises in modern neural networks: double descent

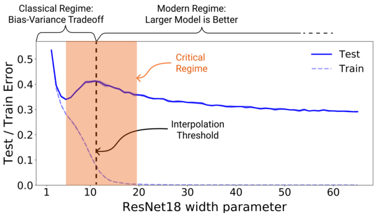



(a) Training and test error on ResNet18 [1]

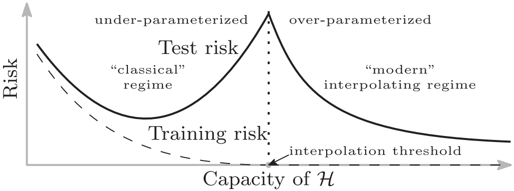



(b) Double descent [2] (Belkin, Hsu, Ma, Mandal, 2019).

# Surprises in modern neural networks: double descent




(a) Training and test error on ResNet18 [1]

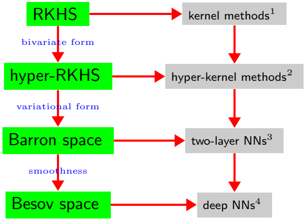


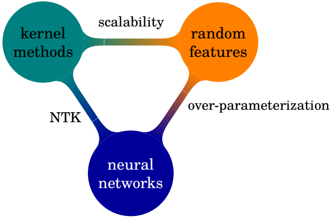

(b) Double descent [2] (Belkin, Hsu, Ma, Mandal, 2019).

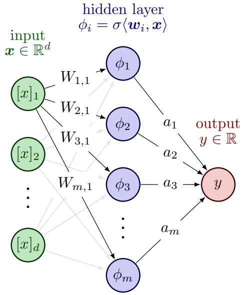
## Observations: beyond bias-variance trade-off

- 1) Monotonic decreasing in the overparameterized regime
- 2) Global minimum when #parameters is infinite
- 3) Peak at the interpolation thresholds

# Today's talk: Function spaces vs. Models (initialization matters)


[1] [LHGYL, JMLR20; LHCS, TPAMI21; LLS, AISTATS21]

[2] [LSHYS, JMLR21]

[3] [LSC, NeurIPS22; LHCS, TPAMI22; LHCS, AISTATS21]

[4] [LVC, NeurIPS22; ZLCC, NeurIPS22; WZLCC, NeurIPS22, ZLCLC, ICML23]

# Questions on high dimensional kernel methods

◦ double descent based on random matrix theory: (Mei and Montanari, 2022), (Hastie, Montanari, Rosset, Tibshirani, 2022), (Liao, Couillet, Mahoney, 2022)

# Questions on high dimensional kernel methods

∘ double descent based on random matrix theory: (Mei and Montanari, 2022), (Hastie, Montanari, Rosset, Tibshirani, 2022), (Liao, Couillet, Mahoney, 2022)

high dimensional kernel methods can only learn linear function! [3]

# Questions on high dimensional kernel methods

∘ double descent based on random matrix theory: (Mei and Montanari, 2022), (Hastie, Montanari, Rosset, Tibshirani, 2022), (Liao, Couillet, Mahoney, 2022)

high dimensional kernel methods can only learn linear function! [3]

∘ asymptotic expansion under high dimensions [4] (El Karoui, 2010)
under the setting of $n, d \to \infty$, $n/d \to \psi_1$ as $d \to \infty$ with $\psi_1 \in (0, \infty)$, we have

$$\|\boldsymbol{K} - (a\boldsymbol{X}\boldsymbol{X}^\top + b\boldsymbol{I})\|_2 \xrightarrow{\mathbb{P}} 0 \quad \text{when } d \to 0 \quad \text{for some parameters } a, b$$

# Questions on high dimensional kernel methods

◦ double descent based on random matrix theory: (Mei and Montanari, 2022), (Hastie, Montanari, Rosset, Tibshirani, 2022), (Liao, Couillet, Mahoney, 2022)

high dimensional kernel methods can only learn linear function! [3]

◦ asymptotic expansion under high dimensions [4] (El Karoui, 2010)
under the setting of $n, d \to \infty$, $n/d \to \psi_1$ as $d \to \infty$ with $\psi_1 \in (0, \infty)$, we have

$$\|\boldsymbol{K} - (a\boldsymbol{X}\boldsymbol{X}^\top + b\boldsymbol{I})\|_2 \xrightarrow{\mathbb{P}} 0 \;\; \text{when } d \to 0 \quad \text{for some parameters } a, b$$

◦ $\|f\|_{\mathcal{H}} < \infty$ ?

# Motivation

- high dimension vs. fixed dimension
- from asymptotic to non-asymptotic
- two-layer neural networks trained by SGD

# Motivation

- high dimension vs. fixed dimension
- from asymptotic to non-asymptotic
- two-layer neural networks trained by SGD

∘ Analysis

- dimension-free bound
- multiple randomness sources
  - data sampling, label noise, Gaussian initialization, stochastic gradients

# Motivation

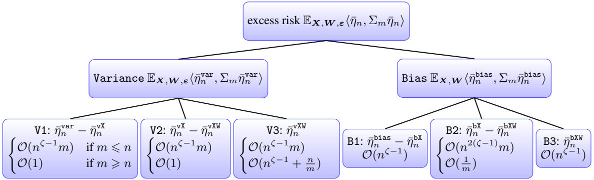
- high dimension vs. fixed dimension
- from asymptotic to non-asymptotic
- two-layer neural networks trained by SGD

○ Analysis

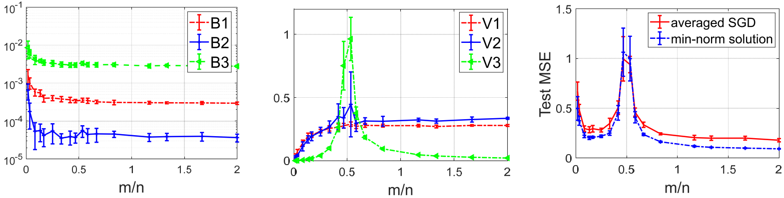
- dimension-free bound
- multiple randomness sources
  - data sampling, label noise, Gaussian initialization, stochastic gradients

observations 1), 2), 3) can be still proved!

# Our results: Double descent of RFMs trained by SGD[1]




- (partially) decouple multiple randomness sources
- converge to $\mathcal{O}(1)$ order (noise variance)
- monotonic decreasing **bias** + unimodal **variance**
- constant step-size SGD does not hurt the convergence rate




(c) bias (d) variance (e) excess risk

[1] **Fanghui Liu**, Johan Suykens, Volkan Cevher. *On the Double Descent of Random Features Models Trained with SGD*. NeurIPS 2022.
**Fanghui Liu**, Xiaolin Huang, Yudong Chen, and Johan Suykens. *Random Features for Kernel Approximation: A Survey on Algorithms, Theory, and Beyond.* TPAMI2021.

# From kernel methods (RKHS) to neural networks (?)

## Kernel Methods

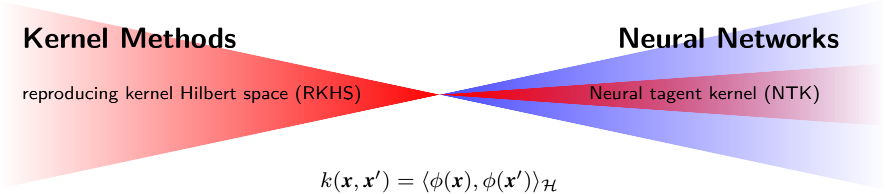
reproducing kernel Hilbert space (RKHS)

## Neural Networks

Neural tagent kernel (NTK)

$$k(\boldsymbol{x}, \boldsymbol{x}') = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle_{\mathcal{H}}$$

# From kernel methods (RKHS) to neural networks (?)




$$k(\boldsymbol{x}, \boldsymbol{x}') = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle_{\mathcal{H}}$$

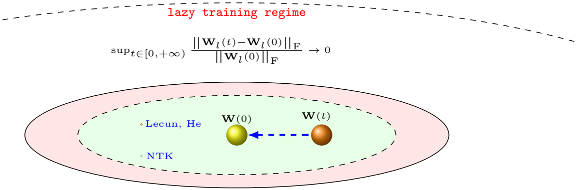
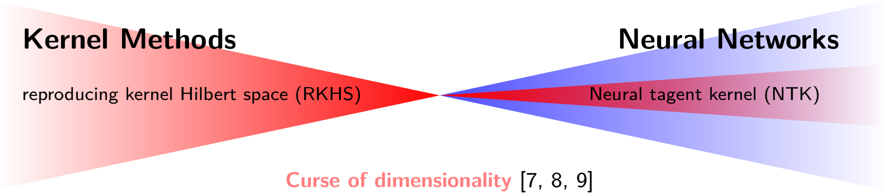


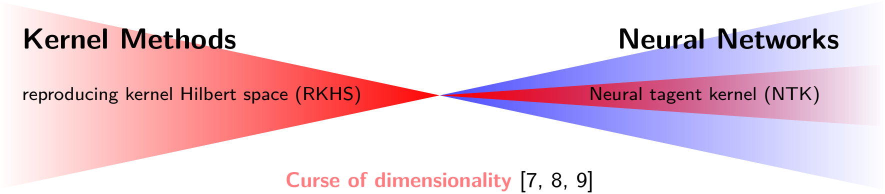

Figure: Lazy training regime: under the NTK initialization [5, 6].

# From kernel methods (RKHS) to neural networks (?)




**Curse of dimensionality** [7, 8, 9]

# From kernel methods (RKHS) to neural networks (?)




Curse of dimensionality [7, 8, 9]

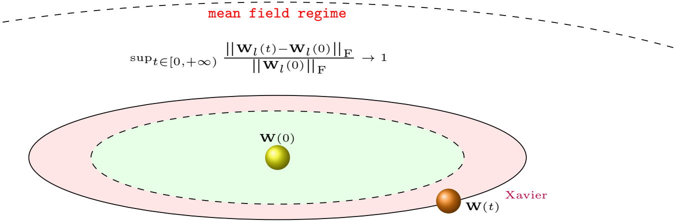



Figure: Mean field regime: under the Xavier initialization, abc-Parametrizations [10, 11].

# From RKHS to Barron space

◦ RKHS of RFMs:

$$\hat{k}_m(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{m} \sum_{i=1}^{m} \phi(\boldsymbol{x}, \boldsymbol{w}_i)\phi(\boldsymbol{x}', \boldsymbol{w}_i) \;\; {\color{red}\rightarrow} \;\; k_\mu(\boldsymbol{x}, \boldsymbol{x}') = \int_{\mathcal{W}} \phi(\boldsymbol{x}, \boldsymbol{w})\phi(\boldsymbol{x}', \boldsymbol{w}) \mathrm{d}\mu(\boldsymbol{w})$$

## From RKHS to Barron space

- RKHS of RFMs:

$$\hat{k}_m(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{m} \sum_{i=1}^{m} \phi(\boldsymbol{x}, \boldsymbol{w}_i)\phi(\boldsymbol{x}', \boldsymbol{w}_i) \rightarrow k_\mu(\boldsymbol{x}, \boldsymbol{x}') = \int_{\mathcal{W}} \phi(\boldsymbol{x}, \boldsymbol{w})\phi(\boldsymbol{x}', \boldsymbol{w}) \mathrm{d}\mu(\boldsymbol{w})$$

**Definition (Barron space [12] (E, Ma, Wu, 2021))**

$$\mathcal{B} = \cup_{\mu \in \mathcal{P}(\mathcal{W})} \mathcal{H}_{k_\mu}, \quad \|f\|_{\mathcal{B}} = \inf_{\mu \in \mathcal{P}(\mathcal{W})} \|f\|_{\mathcal{H}_{k_\mu}}$$

## From RKHS to Barron space

∘ RKHS of RFMs:

$$\hat{k}_m(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{m}\sum_{i=1}^{m} \phi(\boldsymbol{x}, \boldsymbol{w}_i)\phi(\boldsymbol{x}', \boldsymbol{w}_i) \;\rightarrow\; k_\mu(\boldsymbol{x}, \boldsymbol{x}') = \int_{\mathcal{W}} \phi(\boldsymbol{x}, \boldsymbol{w})\phi(\boldsymbol{x}', \boldsymbol{w})\mathrm{d}\mu(\boldsymbol{w})$$

**Definition (Barron space [12] (E, Ma, Wu, 2021))**

$$\mathcal{B} = \cup_{\mu \in \mathcal{P}(\mathcal{W})} \mathcal{H}_{k_\mu}\,, \quad \|f\|_{\mathcal{B}} = \inf_{\mu \in \mathcal{P}(\mathcal{W})} \|f\|_{\mathcal{H}_{k_\mu}}$$

**Remark:** ∘ Two-layer neural networks: data-adaptive kernel

## From RKHS to Barron space

∘ RKHS of RFMs:

$$\hat{k}_m(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{m} \sum_{i=1}^{m} \phi(\boldsymbol{x}, \boldsymbol{w}_i)\phi(\boldsymbol{x}', \boldsymbol{w}_i) \ \rightarrow \ k_\mu(\boldsymbol{x}, \boldsymbol{x}') = \int_{\mathcal{W}} \phi(\boldsymbol{x}, \boldsymbol{w})\phi(\boldsymbol{x}', \boldsymbol{w}) \mathrm{d}\mu(\boldsymbol{w})$$

**Definition (Barron space [12] (E, Ma, Wu, 2021))**

$$\mathcal{B} = \cup_{\mu \in \mathcal{P}(\mathcal{W})} \mathcal{H}_{k_\mu}\,, \quad \|f\|_{\mathcal{B}} = \inf_{\mu \in \mathcal{P}(\mathcal{W})} \|f\|_{\mathcal{H}_{k_\mu}}$$

**Remark:** ∘ Two-layer neural networks: data-adaptive kernel
∘ equivalent to path norm $\|\boldsymbol{\Theta}\|_{\mathcal{P}} := \frac{1}{m} \sum_{k=1}^{m} |a_k| \|\boldsymbol{w}_k\|_1$

## From RKHS to Barron space

∘ RKHS of RFMs:

$$\hat{k}_m(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{m}\sum_{i=1}^{m} \phi(\boldsymbol{x}, \boldsymbol{w}_i)\phi(\boldsymbol{x}', \boldsymbol{w}_i) \;\rightarrow\; k_\mu(\boldsymbol{x}, \boldsymbol{x}') = \int_{\mathcal{W}} \phi(\boldsymbol{x}, \boldsymbol{w})\phi(\boldsymbol{x}', \boldsymbol{w})\mathrm{d}\mu(\boldsymbol{w})$$

**Definition (Barron space [12] (E, Ma, Wu, 2021))**

$$\mathcal{B} = \cup_{\mu \in \mathcal{P}(\mathcal{W})} \mathcal{H}_{k_\mu}\,, \quad \|f\|_{\mathcal{B}} = \inf_{\mu \in \mathcal{P}(\mathcal{W})} \|f\|_{\mathcal{H}_{k_\mu}}$$

**Remark:** ∘ Two-layer neural networks: data-adaptive kernel
∘ equivalent to path norm $\|\boldsymbol{\Theta}\|_{\mathcal{P}} := \frac{1}{m}\sum_{k=1}^{m} |a_k| \|\boldsymbol{w}_k\|_1$
∘ parameter space vs. measure space
e.g., [7] (Bach, 2017), [13] (Bartolucci, Vito, Rosasco, Vigogna, 2022).

# Our results: Refined analyses in Barron spaces[2]

For the class of two-layer neural networks $\mathcal{F}_m$

$$\boldsymbol{\theta}^\star = \operatorname*{arg\,min}_{f_{\boldsymbol{\theta}} \in \mathcal{F}_m} \frac{1}{n} \sum_{i=1}^{n} (y_i - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i))^2 + \lambda \|\boldsymbol{\theta}\|_{\mathcal{P}} \,.$$

[2] Fanghui Liu, Leello Dadi, Volkan Cevher. Learning with two-layer, norm-constrained, over-parameterized neural networks. JMLR (under the second-round review)

# Our results: Refined analyses in Barron spaces[2]

For the class of two-layer neural networks $\mathcal{F}_m$

$$\boldsymbol{\theta}^\star = \operatorname*{arg\,min}_{f_{\boldsymbol{\theta}} \in \mathcal{F}_m} \frac{1}{n} \sum_{i=1}^{n} (y_i - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i))^2 + \lambda \|\boldsymbol{\theta}\|_{\mathcal{P}} \,.$$

**Theorem (Informal)**

*Under proper assumptions, for two-layer over-parameterized neural networks, learning in Barron spaces leads to*

$$\left\| f_{\boldsymbol{\theta}^\star} - f_\rho \right\|^2_{L^2_{\rho_X}} \lesssim \lambda + \frac{1}{m} + d^2 n^{-\frac{d+2}{2d+2}} \qquad w.h.p.$$

[2] Fanghui Liu, Leello Dadi, Volkan Cevher. Learning with two-layer, norm-constrained, over-parameterized neural networks. JMLR (under the second-round review)

# Our results: Refined analyses in Barron spaces[2]

For the class of two-layer neural networks $\mathcal{F}_m$

$$\boldsymbol{\theta}^\star = \operatorname*{arg\,min}_{f_{\boldsymbol{\theta}} \in \mathcal{F}_m} \frac{1}{n} \sum_{i=1}^{n} (y_i - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i))^2 + \lambda \|\boldsymbol{\theta}\|_{\mathcal{P}} \,.$$

## Theorem (Informal)

*Under proper assumptions, for two-layer over-parameterized neural networks, learning in Barron spaces leads to*

$$\left\| f_{\boldsymbol{\theta}^\star} - f_\rho \right\|_{L^2_{\rho_X}}^2 \lesssim \lambda + \frac{1}{m} + d^2 n^{-\frac{d+2}{2d+2}} \qquad w.h.p.$$

**Remark:**

- [14] (Siegel, Xu, 2022) on metric entropy

$$\epsilon^{-\frac{2d}{d+3}} \lesssim_d \log \mathcal{N}_2(\mathcal{G}_1, \epsilon) \lesssim_d \epsilon^{-\frac{2d}{d+3}} \,.$$

[2] Fanghui Liu, Leello Dadi, Volkan Cevher. Learning with two-layer, norm-constrained, over-parameterized neural networks. JMLR (under the second-round review)

# Our results: Refined analyses in Barron spaces[2]

For the class of two-layer neural networks $\mathcal{F}_m$

$$\boldsymbol{\theta}^\star = \mathop{\arg\min}_{f_{\boldsymbol{\theta}} \in \mathcal{F}_m} \frac{1}{n} \sum_{i=1}^{n} (y_i - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i))^2 + \lambda \|\boldsymbol{\theta}\|_{\mathcal{P}} \,.$$

## Theorem (Informal)

*Under proper assumptions, for two-layer over-parameterized neural networks, learning in Barron spaces leads to*

$$\left\| f_{\boldsymbol{\theta}^\star} - f_\rho \right\|^2_{L^2_{\rho_X}} \lesssim \lambda + \frac{1}{m} + d^2 n^{-\frac{d+2}{2d+2}} \qquad w.h.p.$$

**Remark:**

- [14] (Siegel, Xu, 2022) on metric entropy

$$\epsilon^{-\frac{2d}{d+3}}{}_d \lesssim \log \mathcal{N}_2(\mathcal{G}_1, \epsilon) \lesssim_d \epsilon^{-\frac{2d}{d+3}} \leqslant 6144 d^5 \epsilon^{-\frac{2d}{d+2}} \quad \text{[Ours]}$$

[2] Fanghui Liu, Leello Dadi, Volkan Cevher. Learning with two-layer, norm-constrained, over-parameterized neural networks. JMLR (under the second-round review)

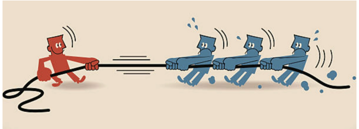
Optimization in Barron spaces is difficult: curse of dimensionality!

Optimization in Barron spaces is difficult: curse of dimensionality!


What is the suitable function space of NNs, both statistically and computationally efficient?

# Applications: Over-parameterization helps/hurts robustness?[3]

**Helps!** [15]

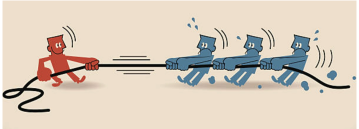


**Hurts!** [16, 17, 18]

[3] Zhenyu Zhu, **Fanghui Liu**, Grigorios Chrysos, Volkan Cevher. *Robustness in deep learning: The good (width), the bad (depth), and the ugly (initialization)*. NeurIPS 2022.
Jiayuan Ye, Zhenyu Zhu, **Fanghui Liu**, Reza Shokri, Volkan Cevher. Initialization matters: Privacy-utility analysis of overparameterized neural networks. NeurIPS 2023.

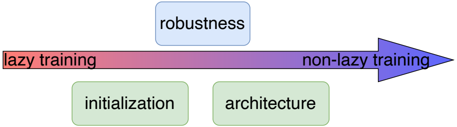
## Applications: Over-parameterization helps/hurts robustness?[3]

**Helps!** [15]



**Hurts!** [16, 17, 18]

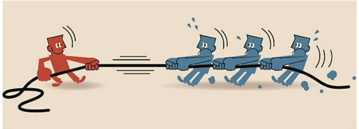
- initialization (e.g., lazy training, non-lazy training)
- architecture (e.g., width, depth)




[3] Zhenyu Zhu, **Fanghui Liu**, Grigorios Chrysos, Volkan Cevher. *Robustness in deep learning: The good (width), the bad (depth), and the ugly (initialization)*. NeurIPS 2022.
Jiayuan Ye, Zhenyu Zhu, **Fanghui Liu**, Reza Shokri, Volkan Cevher. Initialization matters: Privacy-utility analysis of overparameterized neural networks. NeurIPS 2023.

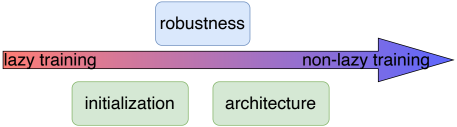
# Applications: Over-parameterization helps/hurts robustness?[3]

**Helps!** [15]



**Hurts!** [16, 17, 18]

- initialization (e.g., lazy training, non-lazy training)
- architecture (e.g., width, depth)

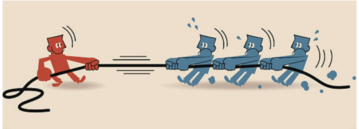



Takeaway messages: **the good (width), the bad (depth), the ugly (initialization)**

[3] Zhenyu Zhu, **Fanghui Liu**, Grigorios Chrysos, Volkan Cevher. *Robustness in deep learning: The good (width), the bad (depth), and the ugly (initialization)*. NeurIPS 2022.
Jiayuan Ye, Zhenyu Zhu, **Fanghui Liu**, Reza Shokri, Volkan Cevher. Initialization matters: Privacy-utility analysis of overparameterized neural networks. NeurIPS 2023.

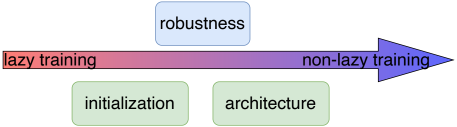
## Applications: Over-parameterization helps/hurts robustness?[3]

**Helps!** [15]



**Hurts!** [16, 17, 18]

- initialization (e.g., lazy training, non-lazy training)
- architecture (e.g., width, depth)




Takeaway messages: **the good (width), the bad (depth), the ugly (initialization)**

- width **helps** robustness in the over-parameterized regime

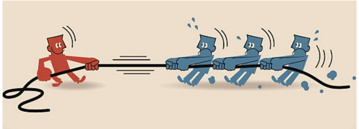
---

[3] Zhenyu Zhu, **Fanghui Liu**, Grigorios Chrysos, Volkan Cevher. *Robustness in deep learning: The good (width), the bad (depth), and the ugly (initialization)*. NeurIPS 2022.

Jiayuan Ye, Zhenyu Zhu, **Fanghui Liu**, Reza Shokri, Volkan Cevher. Initialization matters: Privacy-utility analysis of overparameterized neural networks. NeurIPS 2023.

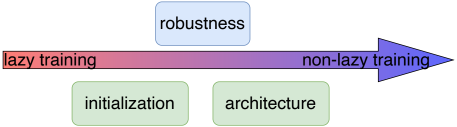
# Applications: Over-parameterization helps/hurts robustness?[3]

**Helps!** [15]



**Hurts!** [16, 17, 18]

- initialization (e.g., lazy training, non-lazy training)
- architecture (e.g., width, depth)



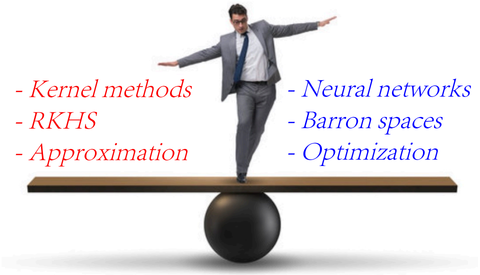

Takeaway messages: **the good (width), the bad (depth), the ugly (initialization)**

- width **helps** robustness in the over-parameterized regime
- depth **helps** robustness in LeCun initialization but **hurts** robustness in He/NTK initialization

[3] Zhenyu Zhu, **Fanghui Liu**, Grigorios Chrysos, Volkan Cevher. *Robustness in deep learning: The good (width), the bad (depth), and the ugly (initialization)*. NeurIPS 2022.

Jiayuan Ye, Zhenyu Zhu, **Fanghui Liu**, Reza Shokri, Volkan Cevher. Initialization matters: Privacy-utility analysis of overparameterized neural networks. NeurIPS 2023.

# Conclusion: the good, the bad, the ugly




| | good | bad | ugly |
|---|---|---|---|
| kernel methods | analysis | performance | curse of dimensionality |
| neural networks | performance | analysis | over-parameterization |
| generalization | benign overfitting | catastrophic overfitting | model complexity |
| robustness | width | depth | initialization |
| privacy | depth | width | initialization |

- IEEE ICASSP 2023 Tutorial - “Neural networks: the good, the bad, and the ugly”
- CVPR 2023 Tutorial - “Deep learning theory for computer vision”

# Thanks for your attention!

# Q & A

my homepage www.lfhsgre.org for more information!

# References I

[1] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever.
Deep double descent: Where bigger models and more data hurt.
In *International Conference on Learning Representations*, 2019.
(Cited on pages 3 and 4.)

[2] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal.
Reconciling modern machine-learning practice and the classical bias–variance trade-off.
*the National Academy of Sciences*, 116(32):15849–15854, 2019.
(Cited on pages 3 and 4.)

[3] Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari.
Linearized two-layers neural networks in high dimension.
*Annals of Statistics*, 49(2):1029–1054, 2021.
(Cited on pages 6, 7, 8, and 9.)

[4] Noureddine El Karoui.
The spectrum of kernel random matrices.
*Annals of Statistics*, 38(1):1–50, 2010.
(Cited on pages 6, 7, 8, and 9.)

# References II


[5] Arthur Jacot, Franck Gabriel, and Clément Hongler.
Neural tangent kernel: Convergence and generalization in neural networks.
In *Advances in Neural Information Processing Systems*, pages 8571–8580, 2018.
(Cited on pages 14 and 15.)

[6] Lenaic Chizat, Edouard Oyallon, and Francis Bach.
On lazy training in differentiable programming.
In *Advances in Neural Information Processing Systems*, pages 2933–2943, 2019.
(Cited on pages 14 and 15.)

[7] Francis Bach.
Breaking the curse of dimensionality with convex neural networks.
*Journal of Machine Learning Research*, 18(1):629–681, 2017.
(Cited on pages 16, 17, 18, 19, 20, 21, and 22.)

[8] Gilad Yehudai and Ohad Shamir.
On the power and limitations of random features for understanding neural networks.
In *Advances in Neural Information Processing Systems*, pages 6594–6604, 2019.
(Cited on pages 16 and 17.)

# References III

[9] Michael Celentano, Theodor Misiakiewicz, and Andrea Montanari.
Minimum complexity interpolation in random features models.
*arXiv preprint arXiv:2103.15996*, 2021.
(Cited on pages 16 and 17.)

[10] Greg Yang and Edward J Hu.
Feature learning in infinite-width neural networks.
*arXiv preprint arXiv:2011.14522*, 2020.
(Cited on pages 16 and 17.)

[11] Lénaïc Chizat and Francis Bach.
On the global convergence of gradient descent for over-parameterized models using optimal transport.
*Advances in Neural Information Processing Systems*, 31:3036–3046, 2018.
(Cited on pages 16 and 17.)

[12] Weinan E, Chao Ma, and Lei Wu.
The barron space and the flow-induced function spaces for neural network models.
*Constructive Approximation*, pages 1–38, 2021.
(Cited on pages 18, 19, 20, 21, and 22.)

# References IV

[13] Francesca Bartolucci, Ernesto De Vito, Lorenzo Rosasco, and Stefano Vigogna.
Understanding neural networks with reproducing kernel Banach spaces.
*Applied and Computational Harmonic Analysis*, 2022.
(Cited on pages 18, 19, 20, 21, and 22.)

[14] Jonathan W Siegel and Jinchao Xu.
Sharp bounds on the approximation rates, metric entropy, and $n$-widths of shallow neural networks.
*arXiv preprint arXiv:2101.12365*, 2021.
(Cited on pages 23, 24, 25, and 26.)

[15] Sébastien Bubeck and Mark Sellke.
A universal law of robustness via isoperimetry.
In *Advances in Neural Information Processing Systems*, pages 28811–28822, 2021.
(Cited on pages 29, 30, 31, 32, and 33.)

[16] Hamed Hassani and Adel Javanmard.
The curse of overparametrization in adversarial training: Precise analysis of robust generalization for random features regression.
*arXiv preprint arXiv:2201.05149*, 2022.
(Cited on pages 29, 30, 31, 32, and 33.)

# References V


[17] Boxi Wu, Jinghui Chen, Deng Cai, Xiaofei He, and Quanquan Gu.
Do wider neural networks really help adversarial robustness?
In *Advances in Neural Information Processing Systems*, pages 7054–7067, 2021.
(Cited on pages 29, 30, 31, 32, and 33.)

[18] Hanxun Huang, Yisen Wang, Sarah Erfani, Quanquan Gu, James Bailey, and Xingjun Ma.
Exploring architectural ingredients of adversarially robust deep neural networks.
In *Advances in Neural Information Processing Systems*, pages 5545–5559, 2021.
(Cited on pages 29, 30, 31, 32, and 33.)