Reconciling modern machine-learning practice and the classical bias-variance trade-off

Benign Overfitting & Double Descent

Zhiquan Liu

July 22, 2025

Nankai University, School of Statistics and Data Science

The Cornerstone: Bias-Variance Decomposition

Goal: Decompose the Expected Prediction Error

We want to analyze the expected squared error of our learned model $\hat{h}(x)$ on a new, unseen data point (x_0, y_0). The expectation is over different training sets $\mathcal{D}$.

$$\text{Error}(x_0) = E_{\mathcal{D},\varepsilon}\left[(y_0 - \hat{h}(x_0))^2\right]$$

where the true model is $y_0 = f(x_0) + \varepsilon$, with $E[\varepsilon] = 0$ and $Var(\varepsilon) = \sigma^2$.

The Cornerstone: Bias-Variance Decomposition

Goal: Decompose the Expected Prediction Error

We want to analyze the expected squared error of our learned model $\hat{h}(x)$ on a new, unseen data point (x_0, y_0). The expectation is over different training sets $\mathcal{D}$.

$$\text{Error}(x_0) = E_{\mathcal{D},\varepsilon}\left[(y_0 - \hat{h}(x_0))^2\right]$$

where the true model is $y_0 = f(x_0) + \varepsilon$, with $E[\varepsilon] = 0$ and $Var(\varepsilon) = \sigma^2$.

Step 1: Substitute the true model

$$\begin{aligned}
\text{Error}(x_0) &= E\left[(f(x_0) + \varepsilon - \hat{h}(x_0))^2\right] \\
&= E\left[(f(x_0) - \hat{h}(x_0))^2 + 2\varepsilon(f(x_0) - \hat{h}(x_0)) + \varepsilon^2\right] \\
&= E\left[(f(x_0) - \hat{h}(x_0))^2\right] + E[2\varepsilon(f(x_0) - \hat{h}(x_0))] + E[\varepsilon^2] \\
&= E\left[(f(x_0) - \hat{h}(x_0))^2\right] + \sigma^2 \quad \text{(Since } f, \hat{h} \text{ are independent of } \varepsilon \text{ and } E[\varepsilon] = 0\text{)}
\end{aligned}$$

Bias-Variance Decomposition (Cont.)

Step 2: Decompose the model error term

Now let's focus on the term $E\left[(f(x_0) - \hat{h}(x_0))^2\right]$. We add and subtract the mean prediction $E[\hat{h}(x_0)]$:

$$
\begin{aligned}
& E\left[(f(x_0) - E[\hat{h}(x_0)] + E[\hat{h}(x_0)] - \hat{h}(x_0))^2\right] \\
&= E\Big[(f(x_0) - E[\hat{h}(x_0)])^2 + (\hat{h}(x_0) - E[\hat{h}(x_0)])^2 \\
&\quad + 2(f(x_0) - E[\hat{h}(x_0)])(\hat{h}(x_0) - E[\hat{h}(x_0)])\Big]
\end{aligned}
$$

Bias-Variance Decomposition (Cont.)

Step 2: Decompose the model error term

Now let's focus on the term $E\left[(f(x_0) - \hat{h}(x_0))^2\right]$. We add and subtract the mean prediction $E[\hat{h}(x_0)]$:

$$\begin{aligned} & E\left[(f(x_0) - E[\hat{h}(x_0)] + E[\hat{h}(x_0)] - \hat{h}(x_0))^2\right] \\ &= E\Big[(f(x_0) - E[\hat{h}(x_0)])^2 + (\hat{h}(x_0) - E[\hat{h}(x_0)])^2 \\ &\quad + 2(f(x_0) - E[\hat{h}(x_0)])(\hat{h}(x_0) - E[\hat{h}(x_0)])\Big] \end{aligned}$$

Step 3: Analyze the cross term

The expectation of the cross term is zero:

$$\begin{aligned} & E\left[2(f(x_0) - E[\hat{h}(x_0)])(\hat{h}(x_0) - E[\hat{h}(x_0)])\right] \\ &= 2(f(x_0) - E[\hat{h}(x_0)]) \cdot E\left[\hat{h}(x_0) - E[\hat{h}(x_0)]\right] \\ &= 2(f(x_0) - E[\hat{h}(x_0)]) \cdot (E[\hat{h}(x_0)] - E[\hat{h}(x_0)]) = 0 \end{aligned}$$

The Cornerstone: Bias-Variance Trade-off

Proposition

For a true model $y = f(x) + \varepsilon$, and our learned model $\hat{h}(x)$, the Expected Prediction Error at a point x_0 is:

$$\begin{aligned}
\text{EPE}(x_0) &= E\left[(y_0 - \hat{h}(x_0))^2\right] \\
&= \underbrace{\left(E[\hat{h}(x_0)] - f(x_0)\right)^2}_{\text{Bias}^2} + \underbrace{E\left[(\hat{h}(x_0) - E[\hat{h}(x_0)])^2\right]}_{\text{Variance}} + \underbrace{\sigma^2}_{\text{Irreducible Error}} \\
&= \text{Bias}^2 + \text{Variance} + \sigma^2
\end{aligned}$$

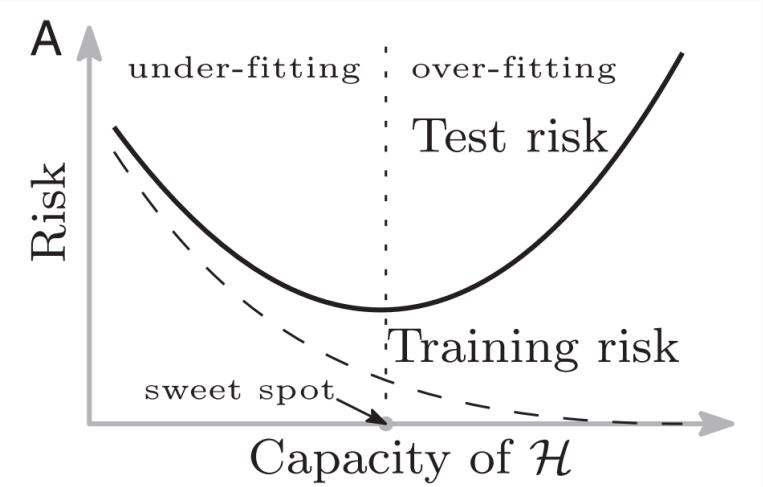
The Cornerstone: Bias-Variance Trade-off

Proposition

For a true model $y = f(x) + \varepsilon$, and our learned model $\hat{h}(x)$, the Expected Prediction Error at a point x_0 is:

$$\begin{aligned}
\text{EPE}(x_0) &= E\left[(y_0 - \hat{h}(x_0))^2\right] \\
&= \underbrace{\left(E[\hat{h}(x_0)] - f(x_0)\right)^2}_{\text{Bias}^2} + \underbrace{E\left[(\hat{h}(x_0) - E[\hat{h}(x_0)])^2\right]}_{\text{Variance}} + \underbrace{\sigma^2}_{\text{Irreducible Error}} \\
&= \text{Bias}^2 + \text{Variance} + \sigma^2
\end{aligned}$$

- **High Bias:** Simple models (e.g., linear) can't capture the true complexity.
- **High Variance:** Complex models (e.g., high-degree polynomial) are too sensitive to the training data.



The Great Divide: Theory vs. Reality

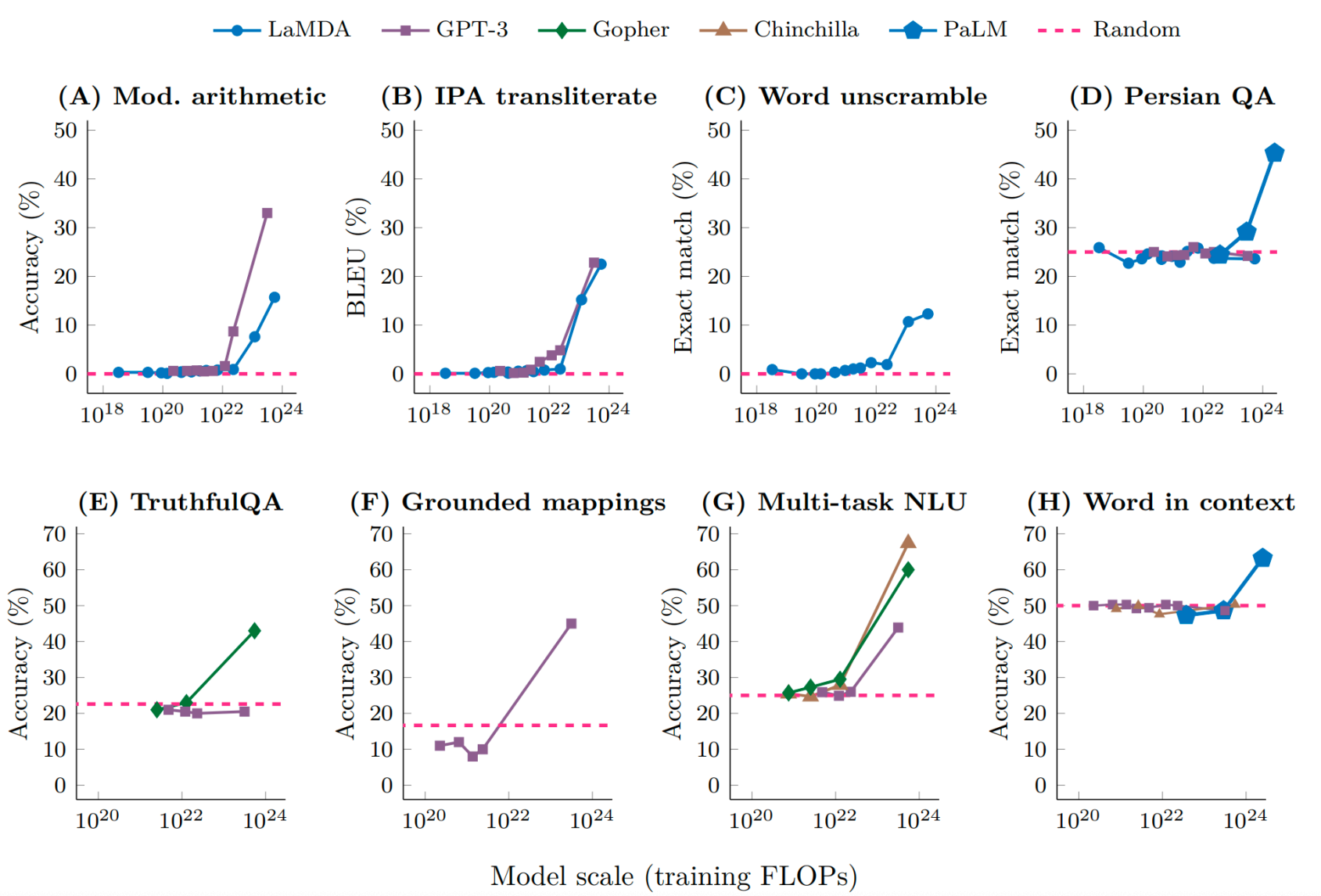


Figure 2: Source: Wei, Tay, Bommasani, et al. (2022), Figure 2.

The Core Question

Is our classical theory flawed?

Why is it that under extreme overparameterization, models not only don't collapse but become even more powerful?

A More Complete Picture: The Double Descent Curve

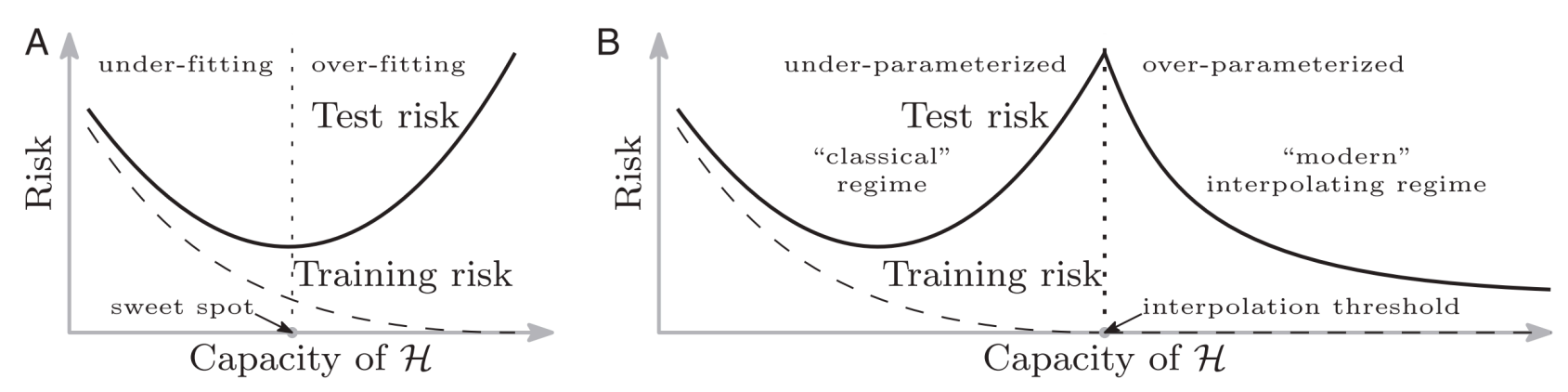


Figure 3: Classical U-shaped curve vs. modern "bigger is better" practice.

Key Regimes

A More Complete Picture: The Double Descent Curve

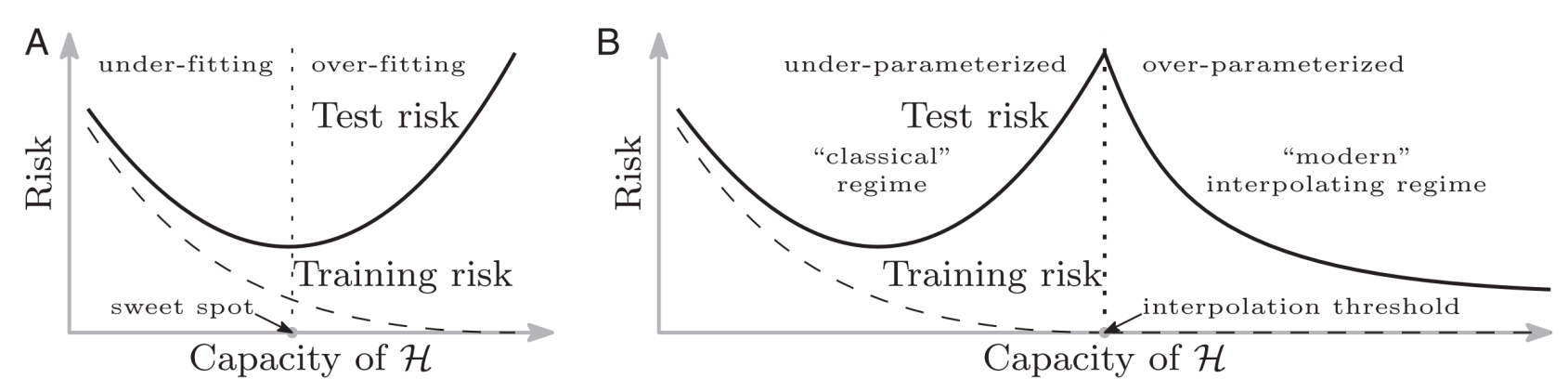


Figure 3: Classical U-shaped curve vs. modern "bigger is better" practice.

Key Regimes

- **Underparameterized ($p < n$):**
  Classical regime. Test error decreases.

A More Complete Picture: The Double Descent Curve

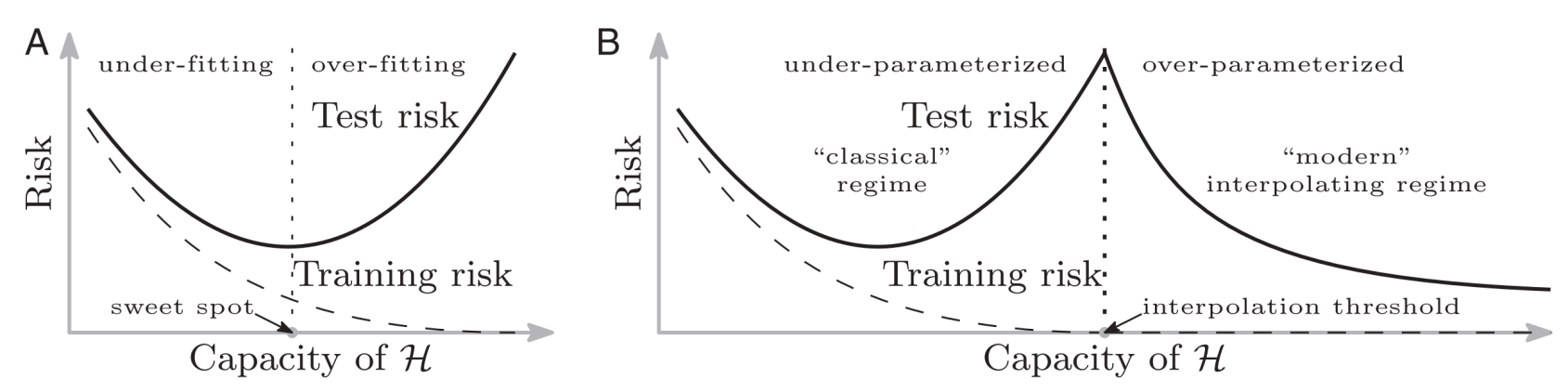


Figure 3: Classical U-shaped curve vs. modern "bigger is better" practice.

Key Regimes

- **Underparameterized ($p < n$):**
  Classical regime. Test error decreases.
- **Interpolation Threshold ($p \approx n$):**
  Critically parameterized. Test error peaks.

A More Complete Picture: The Double Descent Curve

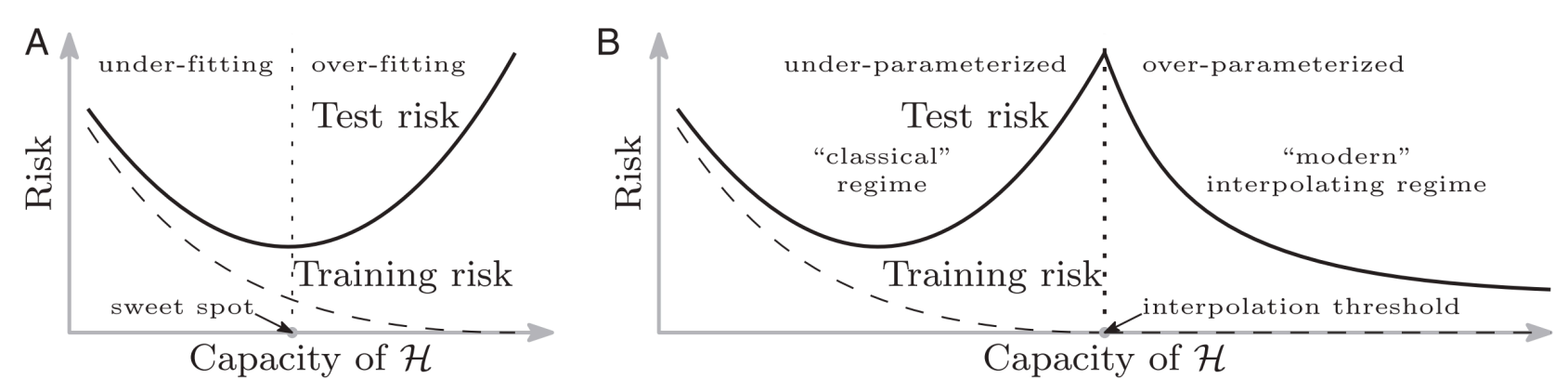


Figure 3: Classical U-shaped curve vs. modern "bigger is better" practice.

Key Regimes

- **Underparameterized ($p < n$):**
  Classical regime. Test error decreases.
- **Interpolation Threshold ($p \approx n$):**
  Critically parameterized. Test error peaks.
- **Overparameterized ($p > n$):**
  Modern regime. Test error descends again.

The Theoretical Underpinning

Approximation Theorem

In a noiseless setting, for **any** function $h \in \mathcal{H}_\infty$ that **interpolates** the data, its uniform error is bounded with high probability:

$$\sup_{x \in \Omega} |h(x) - h^*(x)| \leq A e^{-B(n/\log n)^{1/d}} (\|h^*\|_{\mathcal{H}_\infty} + \|h\|_{\mathcal{H}_\infty})$$

where h^* is the true function and $\mathcal{H}_\infty$ is the function space.

The Theoretical Underpinning

Approximation Theorem

In a noiseless setting, for **any** function $h \in \mathcal{H}_\infty$ that **interpolates** the data, its uniform error is bounded with high probability:

$$\sup_{x \in \Omega} |h(x) - h^*(x)| \leq A e^{-B(n/\log n)^{1/d}} (\|h^*\|_{\mathcal{H}_\infty} + \|h\|_{\mathcal{H}_\infty})$$

where h^* is the true function and $\mathcal{H}_\infty$ is the function space.

The error bound is proportional to the sum of the norms $(\|h^*\| + \|h\|)$. To minimize the error bound, one should choose the interpolating solution h with the **minimum norm**.

Proof Sketch

Approximation Theorem

In a noiseless setting, for **any** function $h \in \mathcal{H}_\infty$ that **interpolates** the data, its uniform error is bounded with high probability:

$$\sup_{x \in \Omega} |h(x) - h^*(x)| \leq A e^{-B(n/\log n)^{1/d}} (\|h^*\|_{\mathcal{H}_\infty} + \|h\|_{\mathcal{H}_\infty})$$

where h^* is the true function and $\mathcal{H}_\infty$ is the function space.

The high-level idea is to bound the error function $f(x) := h(x) - h^*(x)$.

Proof Sketch

Approximation Theorem

In a noiseless setting, for **any** function $h \in \mathcal{H}_\infty$ that **interpolates** the data, its uniform error is bounded with high probability:

$$\sup_{x \in \Omega} |h(x) - h^*(x)| \leq A e^{-B(n/\log n)^{1/d}} (\|h^*\|_{\mathcal{H}_\infty} + \|h\|_{\mathcal{H}_\infty})$$

where h^* is the true function and $\mathcal{H}_\infty$ is the function space.

The high-level idea is to bound the error function $f(x) := h(x) - h^*(x)$.

1. **Define Fill Distance κ_n:** This measures how well the data points $\{x_i\}$ cover the space Ω. By approximation theory, κ_n shrinks as n increases with high probability: $\kappa_n = O((n/\log n)^{-1/d})$.

Proof Sketch

Approximation Theorem

In a noiseless setting, for **any** function $h \in \mathcal{H}_\infty$ that **interpolates** the data, its uniform error is bounded with high probability:

$$\sup_{x \in \Omega} |h(x) - h^*(x)| \leq Ae^{-B(n/\log n)^{1/d}}(\|h^*\|_{\mathcal{H}_\infty} + \|h\|_{\mathcal{H}_\infty})$$

where h^* is the true function and $\mathcal{H}_\infty$ is the function space.

The high-level idea is to bound the error function $f(x) := h(x) - h^*(x)$.

1. **Define Fill Distance κ_n:** This measures how well the data points $\{x_i\}$ cover the space Ω. By approximation theory, κ_n shrinks as n increases with high probability: $\kappa_n = O((n/\log n)^{-1/d})$.
2. **Observe Properties of the Error Function $f(x)$:**
   - Zero on training points: $f(x_i) = h(x_i) - h^*(x_i) = y_i - y_i = 0$.
   - By the triangle inequality: $\|f\|_{\mathcal{H}} \leq \|h\|_{\mathcal{H}} + \|h^*\|_{\mathcal{H}}$.

Proof Sketch

Approximation Theorem

In a noiseless setting, for **any** function $h \in \mathcal{H}_\infty$ that **interpolates** the data, its uniform error is bounded with high probability:

$$\sup_{x \in \Omega} |h(x) - h^*(x)| \leq A e^{-B(n/\log n)^{1/d}} (\|h^*\|_{\mathcal{H}_\infty} + \|h\|_{\mathcal{H}_\infty})$$

where h^* is the true function and $\mathcal{H}_\infty$ is the function space.

The high-level idea is to bound the error function $f(x) := h(x) - h^*(x)$.

1. **Define Fill Distance κ_n:** This measures how well the data points $\{x_i\}$ cover the space Ω. By approximation theory, κ_n shrinks as n increases with high probability: $\kappa_n = O((n/\log n)^{-1/d})$.
2. **Observe Properties of the Error Function $f(x)$:**
   - Zero on training points: $f(x_i) = h(x_i) - h^*(x_i) = y_i - y_i = 0$.
   - By the triangle inequality: $\|f\|_{\mathcal{H}} \leq \|h\|_{\mathcal{H}} + \|h^*\|_{\mathcal{H}}$.
3. **Apply a Known Theorem:** Thm. 11.22(Wendland, 2004) bounds the maximum of such a $f(x)$ using $\|f\|_{\mathcal{H}}$ and κ_n.

Evidence: Random Fourier Features (RFF)

What is RFF?

- It can be viewed as a **simplified 2-layer neural network**.

Evidence: Random Fourier Features (RFF)

What is RFF?

- It can be viewed as a **simplified 2-layer neural network**.
- The first layer weights (v_k) are randomly sampled and then fixed.

Evidence: Random Fourier Features (RFF)

What is RFF?

- It can be viewed as a **simplified 2-layer neural network**.
- The first layer weights (v_k) are randomly sampled and then fixed.
- Only the second layer weights (α_k) are learned during training.

Evidence: Random Fourier Features (RFF)

What is RFF?

- It can be viewed as a **simplified 2-layer neural network**.
- The first layer weights (v_k) are randomly sampled and then fixed.
- Only the second layer weights (α_k) are learned during training.

Mathematical Form
The model function $h(x)$ is a linear combination of N random features:

$$h(x) = \sum_{k=1}^{N} \alpha_k \phi(x; v_k)$$

N : The number of features. The crucial knob to control **model capacity**.

v_k : The fixed, random vectors (e.g., from a Gaussian distribution).

α_k : The learnable coefficients.

The Experimental Setup

Setup Details

- **Dataset:** A subset of MNIST (10-class handwritten digits).
- **Training Size (n): 10,000** samples. This number is the key to locating the interpolation threshold.
- **Learning:** Empirical Risk Minimization (ERM) with Squared Loss.

The Experimental Setup

Setup Details

- **Dataset:** A subset of MNIST (10-class handwritten digits).
- **Training Size (n): 10,000** samples. This number is the key to locating the interpolation threshold.
- **Learning:** Empirical Risk Minimization (ERM) with Squared Loss.

The Crucial Choice: The Implicit Bias

- When $N > n$, there are **infinite** interpolating solutions (solutions with zero training error).

The Experimental Setup

Setup Details

- **Dataset:** A subset of MNIST (10-class handwritten digits).
- **Training Size (n): 10,000** samples. This number is the key to locating the interpolation threshold.
- **Learning:** Empirical Risk Minimization (ERM) with Squared Loss.

The Crucial Choice: The Implicit Bias

- When $N > n$, there are **infinite** interpolating solutions (solutions with zero training error).
- Which one to choose? The authors select the solution with the **minimum L2-norm** of the coefficients $\|\alpha\|_2$.

The Experimental Setup

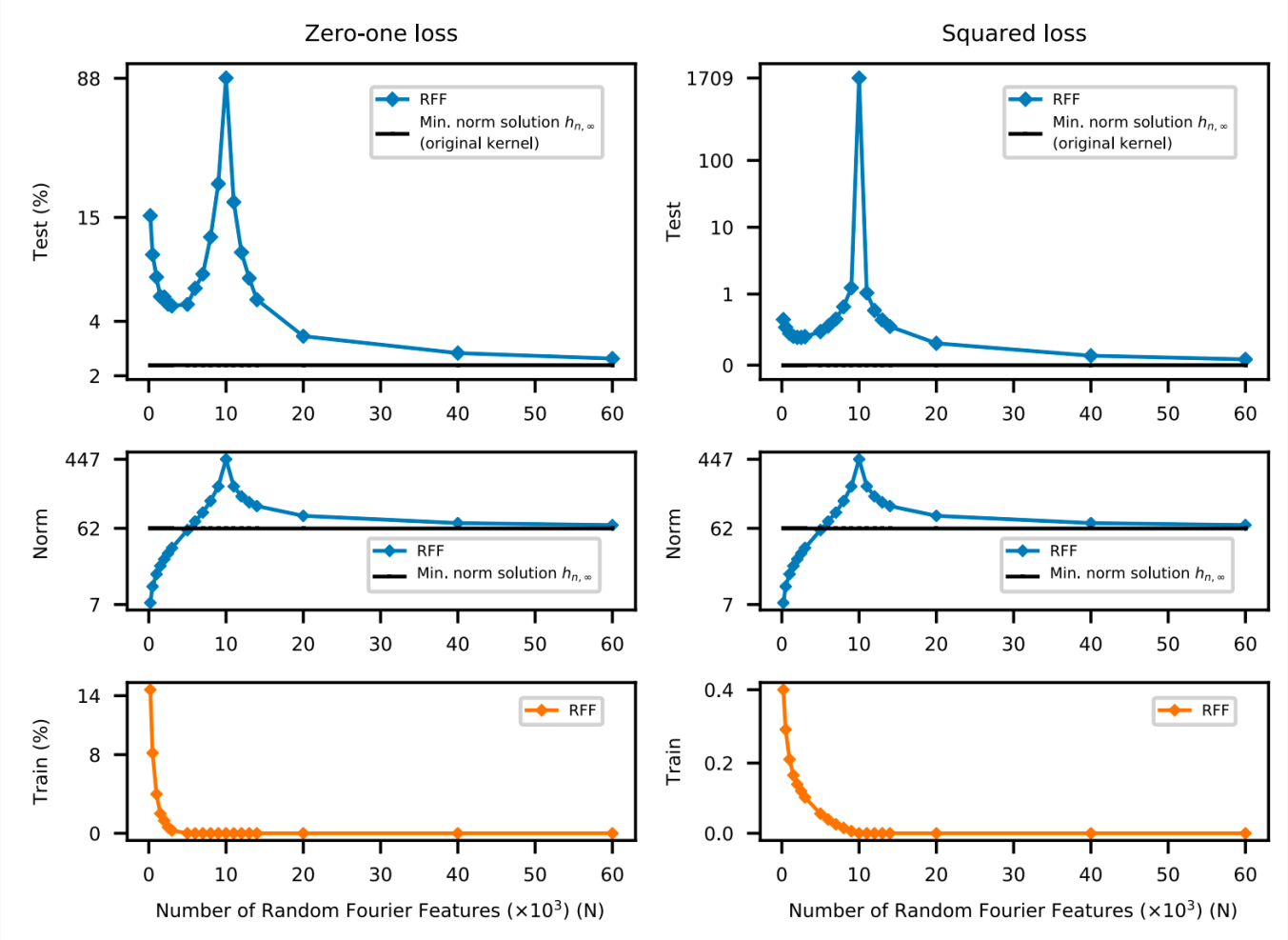
Setup Details

- **Dataset:** A subset of MNIST (10-class handwritten digits).
- **Training Size (n): 10,000** samples. This number is the key to locating the interpolation threshold.
- **Learning:** Empirical Risk Minimization (ERM) with Squared Loss.

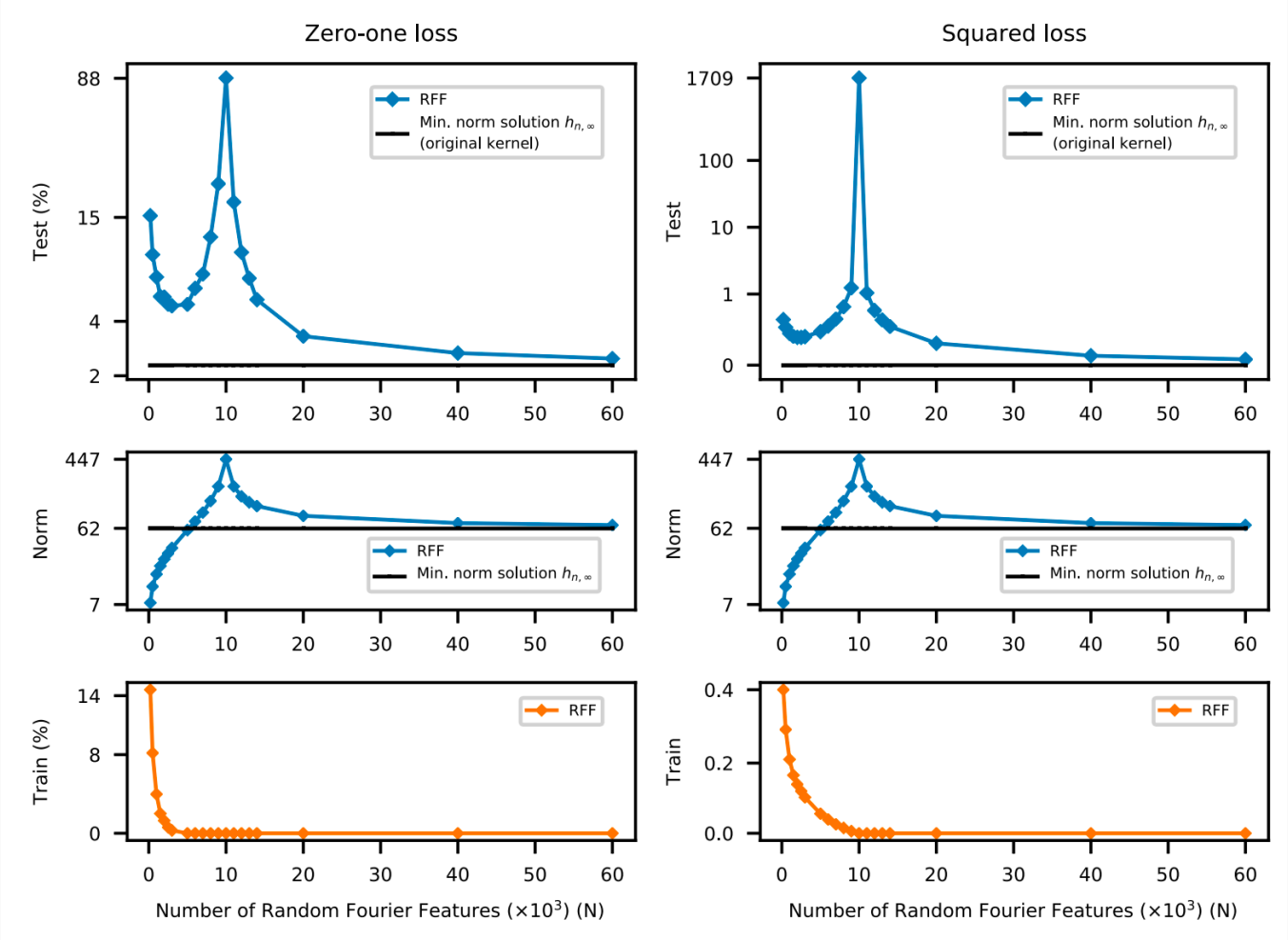
The Crucial Choice: The Implicit Bias

- When $N > n$, there are **infinite** interpolating solutions (solutions with zero training error).
- Which one to choose? The authors select the solution with the **minimum L2-norm** of the coefficients $\|\alpha\|_2$.
- This mimics an **implicit bias** of optimization algorithms (like gradient descent) towards "simpler" or "smoother" functions.

Results from RFF



Results from RFF



In the overparameterized regime, as model capacity increases, the norm of the learned solution **decreases**, correlating with lower test error. This points to an **implicit bias** towards simpler solutions.

From Ideal Theory to Real-World Practice

The theory favors the *minimum norm* solution. But how do we ensure that practical algorithms find it?

From Ideal Theory to Real-World Practice

The theory favors the *minimum norm* solution. But how do we ensure that practical algorithms find it?

The answer lies in the **Inductive Bias** of the algorithm, which can be:

- **Explicit:** In models like **Kernel Machines**, the solution is mathematically defined to be the minimum norm interpolant in an RKHS.

From Ideal Theory to Real-World Practice

The theory favors the *minimum norm* solution. But how do we ensure that practical algorithms find it?

The answer lies in the **Inductive Bias** of the algorithm, which can be:

- **Explicit:** In models like **Kernel Machines**, the solution is mathematically defined to be the minimum norm interpolant in an RKHS.
- **Implicit:** In **Deep Neural Networks**, the bias comes from the **optimization algorithm** (e.g., SGD). While not fully understood, evidence points towards a preference for "simplicity":
  - SGD on linear models converges to the **max-margin** solution.
  - In deep networks, SGD often leads to **Neural Collapse** (Papyan et al., 2020), a highly structured and simple geometric configuration of features.
  - Minimum norm is just one well-studied instance of this broader principle.

Summary

1. The Classical Theory is Incomplete, Not Wrong

The U-shaped curve describes the underparameterized regime, not the whole story.

2. "Double Descent" Reveals a New "Bigger is Better" Paradigm

Increasing model capacity beyond the interpolation threshold can surprisingly improve generalization.

3. The Secret Sauce: Algorithm's Implicit Bias towards Simplicity

Overparameterization creates infinite solutions. The optimization algorithm's implicit bias (e.g., towards minimum norm) acts as an invisible regularizer, selecting the one that generalizes well.

The Grand Unifying Principle

Overparameterization + Implicit Bias = Benign Overfitting

Thank you.

Any questions?