

## The Deep Learning Crisis

**The Context:** Deep Networks are heavily over-parameterized, and fit the training data perfectly while maintaining the ability to generalize.

*Zhang et al. (2017)* showed DNNs fit **random labels** perfectly, these NNs have the brute force capacity to memorize noise..

**The Theoretical Crisis:** Classical theory (VC Dimension, Bias-Variance) predicts that:

High Capacity + Zero Training Error  $\Rightarrow$  Massive Overfitting

**But in Deep Learning, this is FALSE.**

**The Hypothesis:** How are DNNs able to learn even when interpolating training data ?

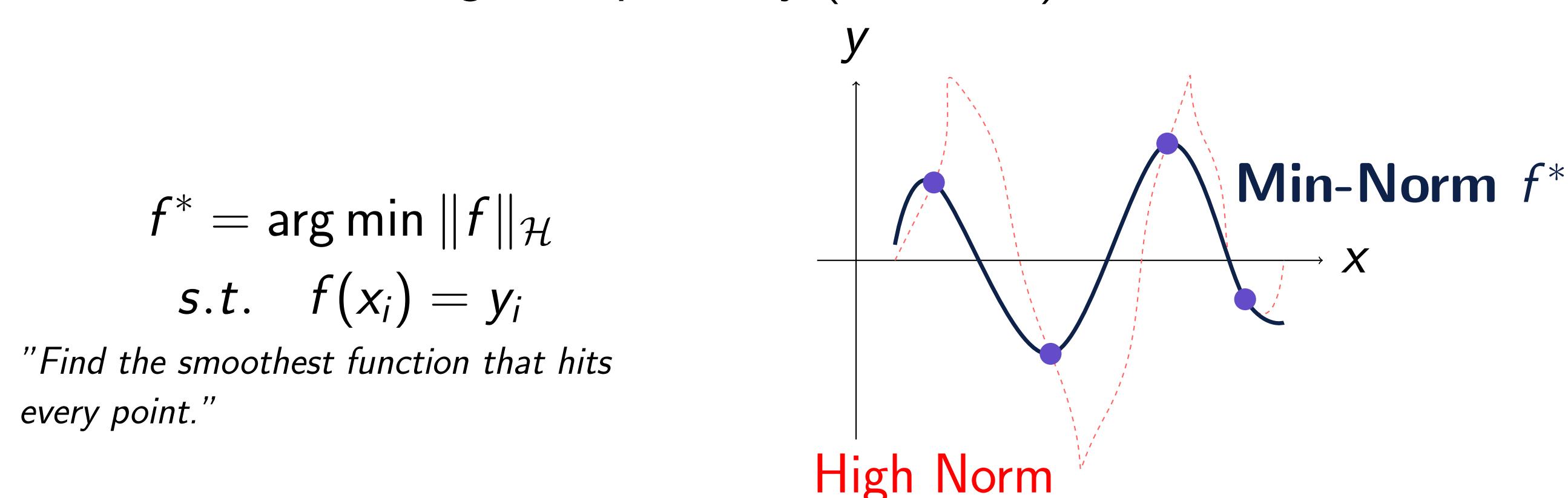
This paper shows that interpolation also applies to classical **Kernel Machines** and proposes to find the missing theory by studying kernel learning.

## Kernel Learning Basics

- **The Idea:** Map low-dimensional data  $x$  to a high-dimensional (infinite) feature space  $\phi(x)$  to make it linearly separable.
- **Kernel Trick:** We define a function  $K(x, z) = \langle \phi(x), \phi(z) \rangle$  to compute distances without visiting the infinite space explicitly.
- **RKHS:** The "Reproducing Kernel Hilbert Space" ( $\mathcal{H}$ ) is the space of all possible functions built from this kernel.

**What is the relation with DL ?**

- Kernel Machines are essentially **Infinite-Width Two-Layer Neural Networks** with a fixed first layer (*Jacot et al. (2018)*)
- **Min-Norm Interpolant:** We analyze the "simplest" function ( $f^*$ ) that fits training data perfectly ( $Loss = 0$ ).



## Why Theory Fails

**Theorem 1:** In a noisy label set-up, fitting  $n$  points exactly with a smooth kernel forces the norm to grow exponentially:

$$\|f^*\|_{\mathcal{H}} \approx O(e^n)$$

Under previous theory, generalisation bounds depend linearly on  $\|f^*\|_{\mathcal{H}}$  and so become trivial. Previous theory cannot explain the performance of interpolating kernels.

## "Overfitting" is a Myth

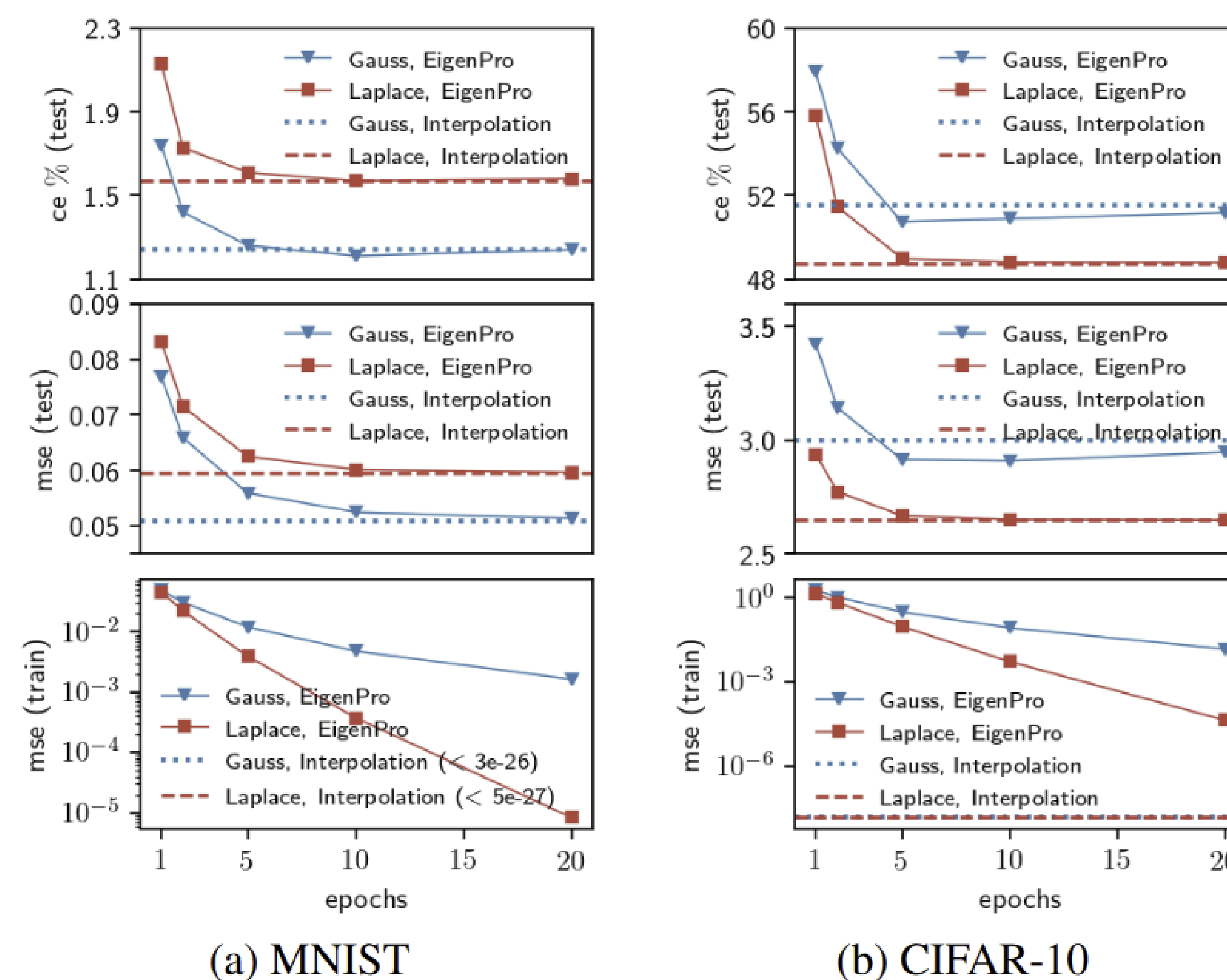


Figure: Comparison of approximate classifiers trained by EigenPro-SGD and interpolated classifiers

**Key Parallel:** This mirrors Deep Networks trained with SGD, continued training (driving loss to 0) allows the model to "settle" into a good solution (Double Descent).

## The ReLU vs. Laplacian Connection

*Zhang et al.* showed ReLU NNs can fit random noise. The authors test with different kernel types:

Model Type	Structure	Fit Random Labels?
ReLU Networks	Non-Smooth (Kinks)	YES (Fast)
Laplacian Kernel	Non-Smooth (Spiky)	YES (Fast)
Gaussian Kernel	Very Smooth	NO (Slow)

- The "spiky" nature of the Laplacian kernel mimics the **ReLU activation function**. Both are optimization-friendly for fitting noise, unlike smooth Gaussian kernels.
- Once the kernels are overfit, both show very similar classification and regression performance on test data.

**Conclusion:** Optimization speed depends on geometry (Smooth vs Spiky), but **Generalization depends on the Norm**.

## The Role of SGD

Why do massive networks find the "simple" solution?

- **In Kernels:** SGD initialized at 0 or in the span of centered kernels converges to the *Minimum Norm Solution*.
- **In Deep Nets:** Evidence suggests SGD induces a similar "implicit regularization."

**Key Takeaway:** It is not the *architecture* preventing overfitting, it is the *algorithm* (SGD) selecting a specific low-complexity solution.

## Robustness to Label Noise

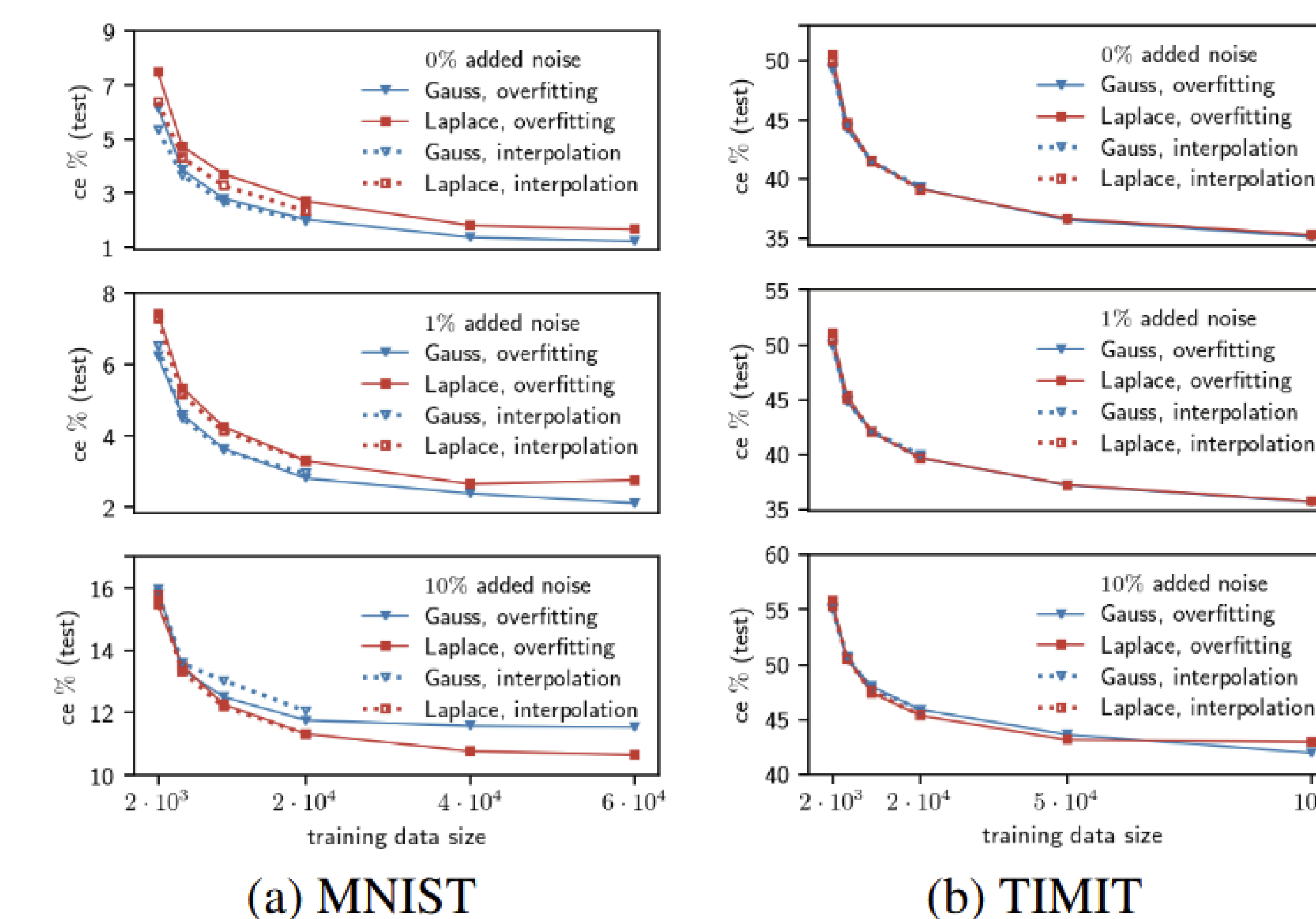


Figure 6: Overfitted and interpolated classifiers using Gaussian kernel and Laplace kernel for datasets with added label noise (top: 0%, middle: 1%, bottom: 10%)

Figure: Performance on Corrupted Labels

Even with 10% corrupted labels, the Kernel Machine fits the noise (bad locally) but maintains near-optimal test error (good globally).

## Conclusion for DL

- 1 **Universality:** The "Generalization Puzzle" is a property of *interpolating high-dimensional data*, not just Deep Learning.
- 2 **Inductive Bias is King:** Success is likely driven by the **inductive bias** (minimum norm).
- 3 **Optimization  $\neq$  Generalization:** Laplacian and Gaussian kernels optimize differently (like ReLU vs Tanh) but generalize similarly.