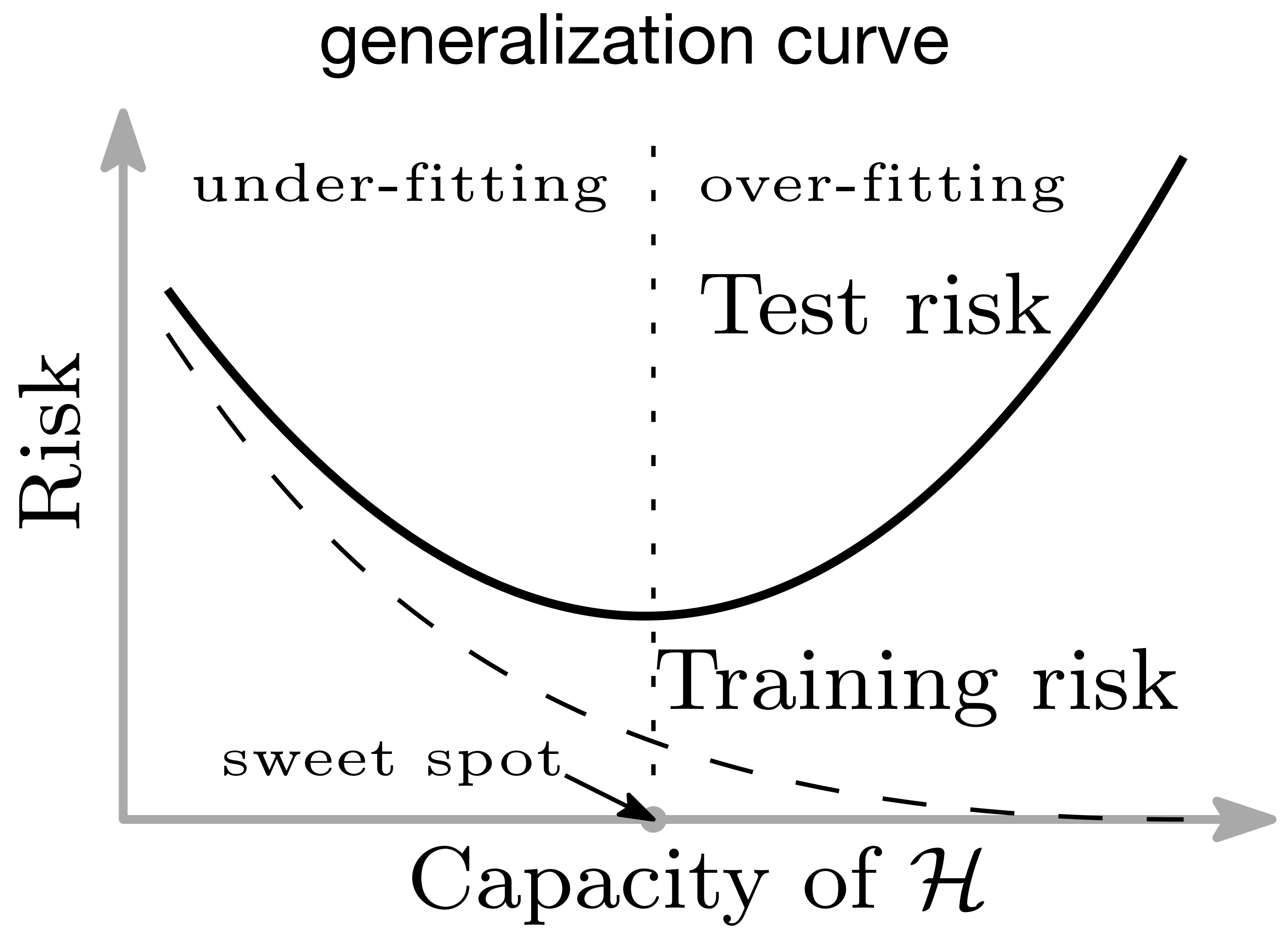
A U-turn on Double Descent:
Rethinking Parameter Counting in Statistical Learning
(NeurIPS 2023 Oral presentation)

OptiML Group Meeting
March 7, 2024

Presented by Hanseul Cho

OptiML
Optimization & Machine Learning

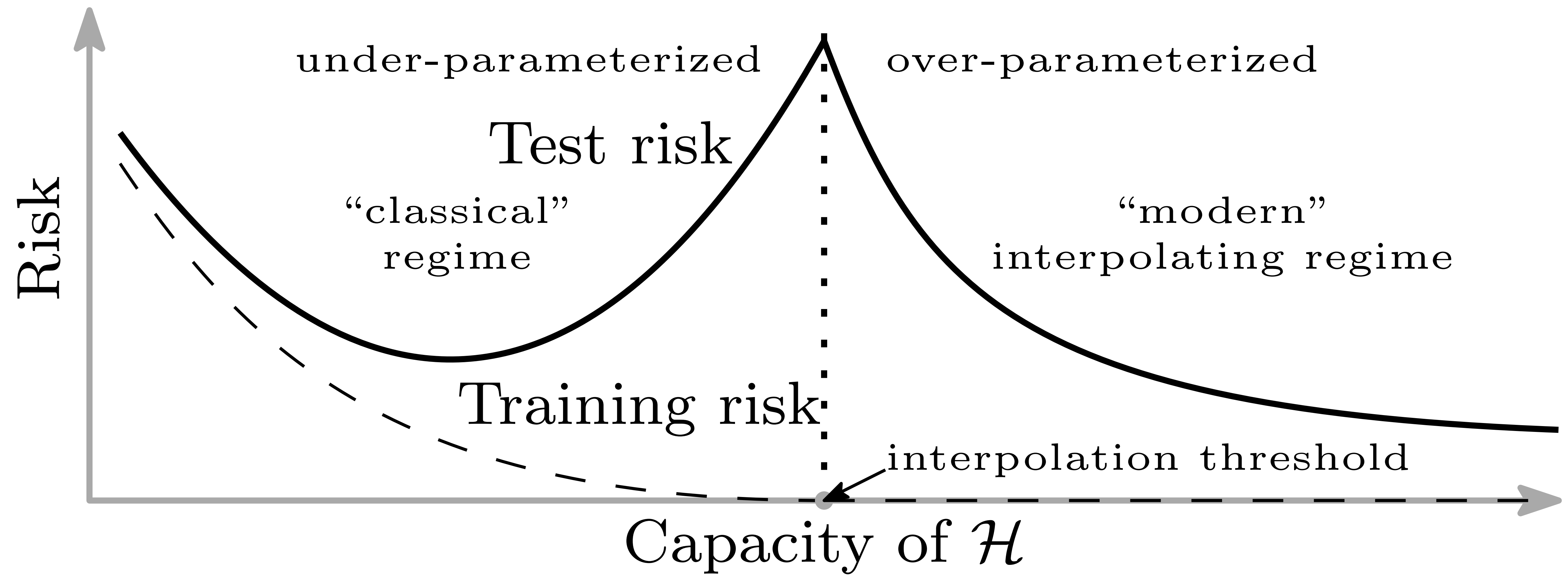
Background: Model size vs. Test error



- Classical theory on the relationship btw model complexity & prediction error
- Under-fitting: Low model capacity, High bias
- Over-fitting: High model capacity, High variance

Background: Double descent

generalization curve



- Belkin et al. [BHMM19] :
  - “Double Descent” happens if the total # of params P FURTHER grows.
  - “Interpolation regime”: # of data $n < P$ & train-error = 0.

Overview

Q. But how is the model complexity being computed?

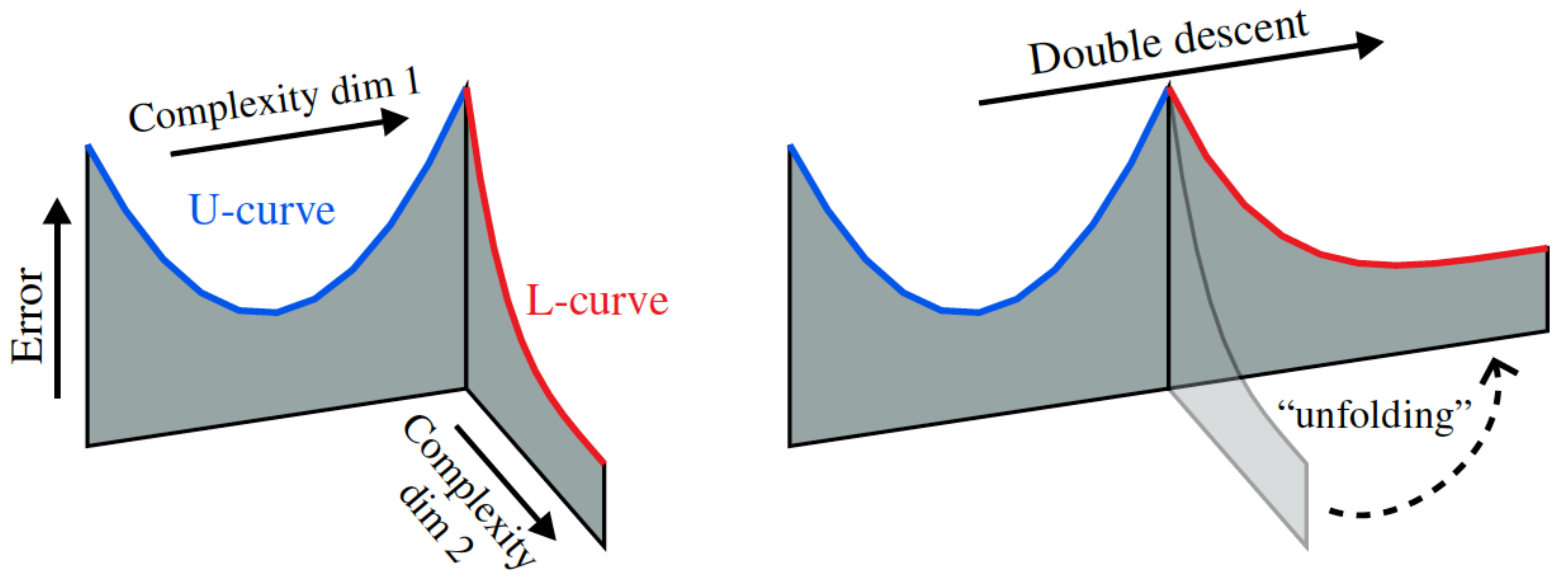
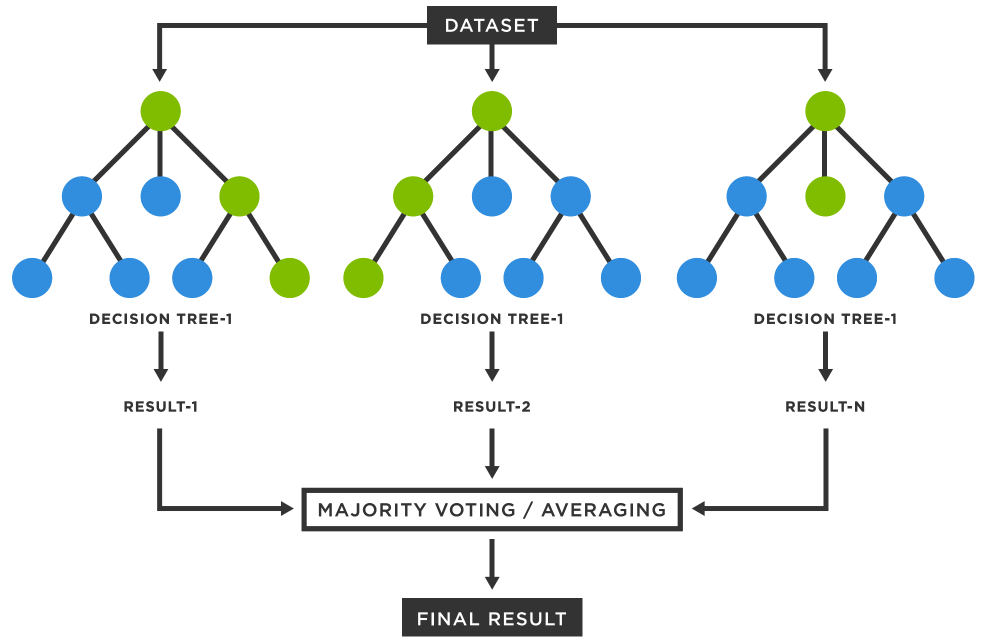


Figure 1: **A 3D generalization plot with two complexity axes unfolding into double descent.** A generalization plot with two complexity axes, each exhibiting a convex curve (left). By increasing raw parameters along different axes sequentially, a double descent effect appears to emerge along their composite axis (right).

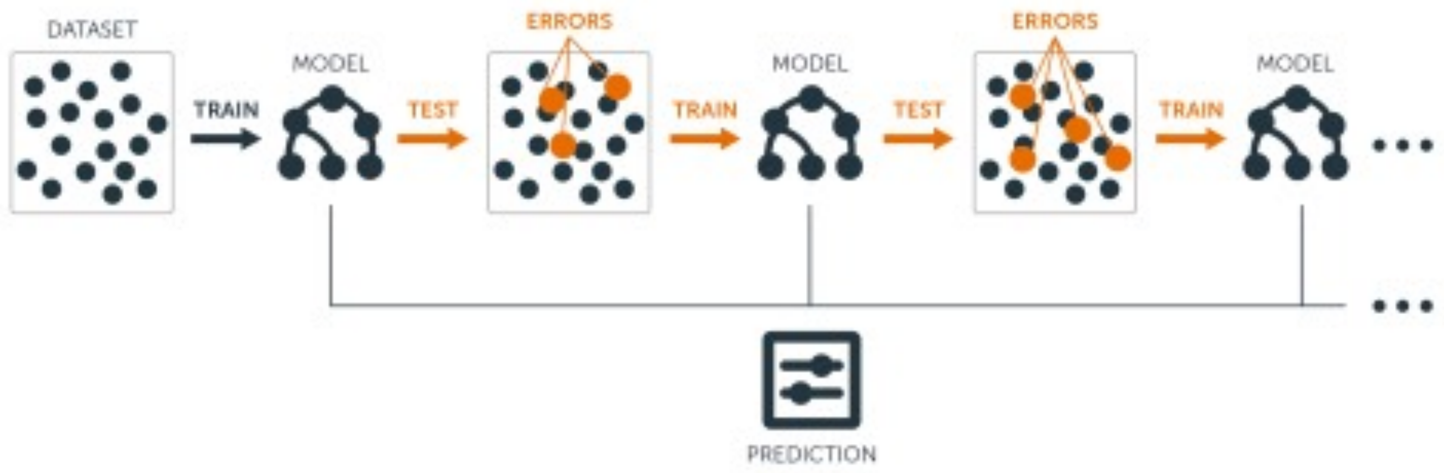
- For non-deep ML methods, the double descent phenomenon can be explained under existing paradigms by <u>rethinking the parameter counting</u> [CJvdS23].

Experimental setup

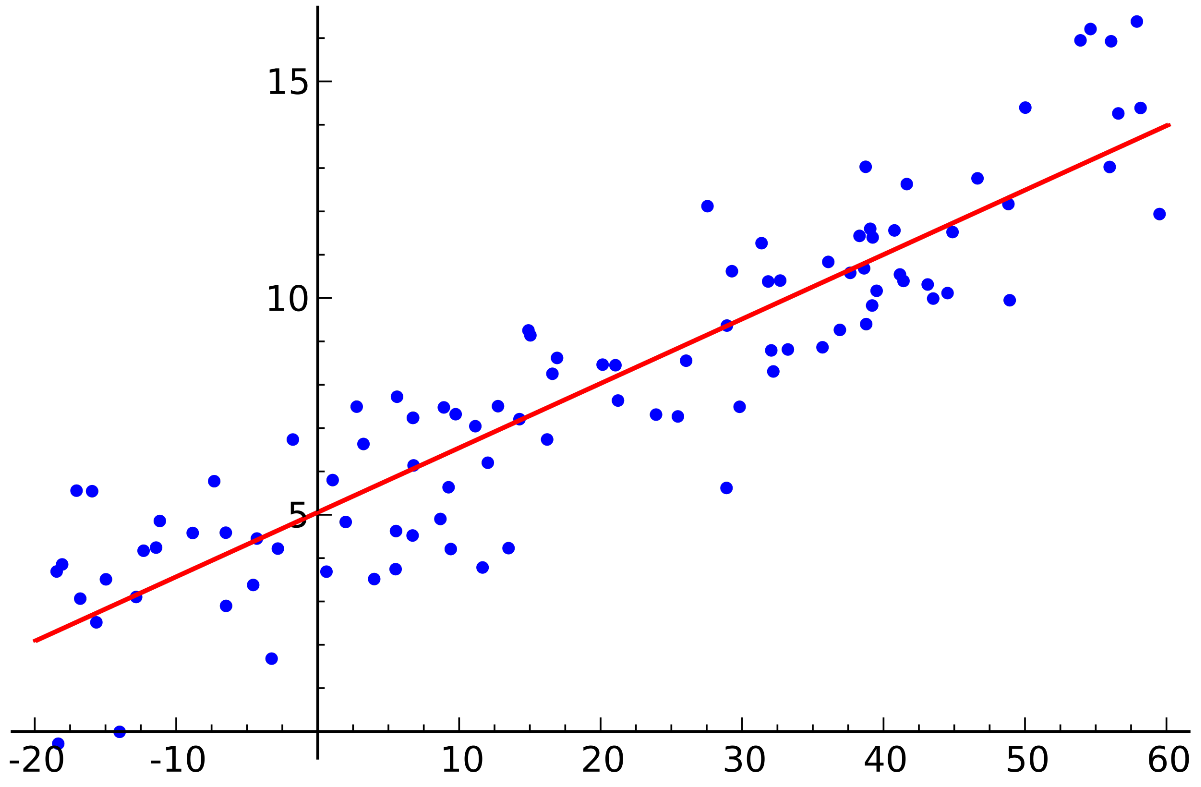
- Non-deep ML methods: trees, gradient boosting, and linear regressions. [BHMM19]



https://www.spotfire.com/glossary/what-is-a-random-forest



https://machine-learning.paperspace.com/wiki/gradient-boosting



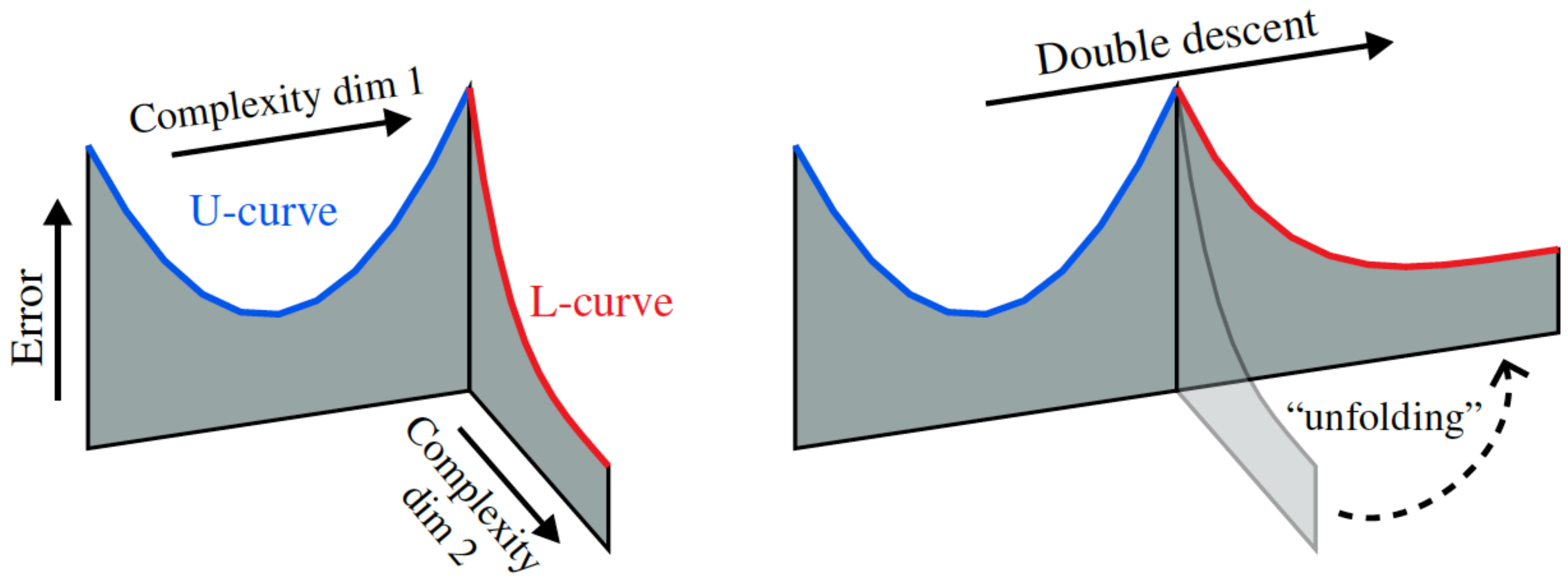
https://en.wikipedia.org/wiki/Linear_regression

- Classification of MNIST ($n = 10000$)

- Minimizing the squared loss / One-vs-rest strategy

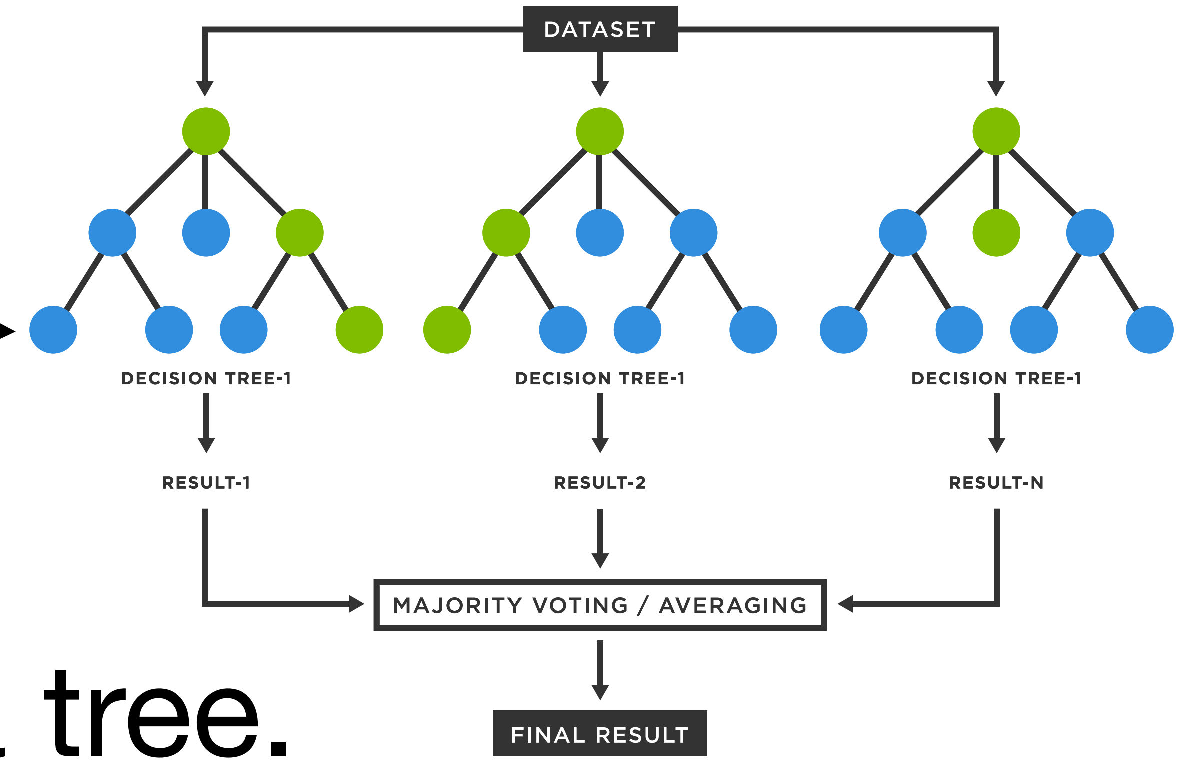
Part 1. Revisiting existing results [BHMM19]

Revisiting the evidence for double descent in non-deep ML models

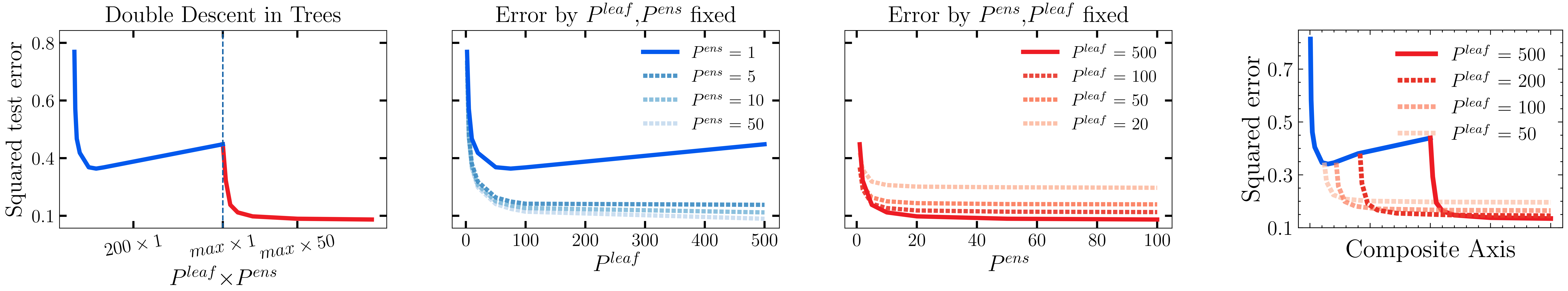


- “There’s *more than one* complexity axis along which the param count grows.”
- “The location of the second descent is not tied to the interpolation threshold ($P = n$).”

Double descent in trees 🌳



- P^{leaf}: maximum allowed number of terminal leaf nodes of a tree.
- ⚠️ $P^{leaf} \leq n$ (when every leaf contains only one instance)
- To further increase the model complexity, [BHMM19] manipulate:
- P^{ens}: number of different trees grown to full depth. → multiple trees.



Double descent in gradient boosting 💨

The gradient boosted trees algorithm. We consider gradient boosting with learning rate η and the squared loss as $L(\cdot, \cdot)$:

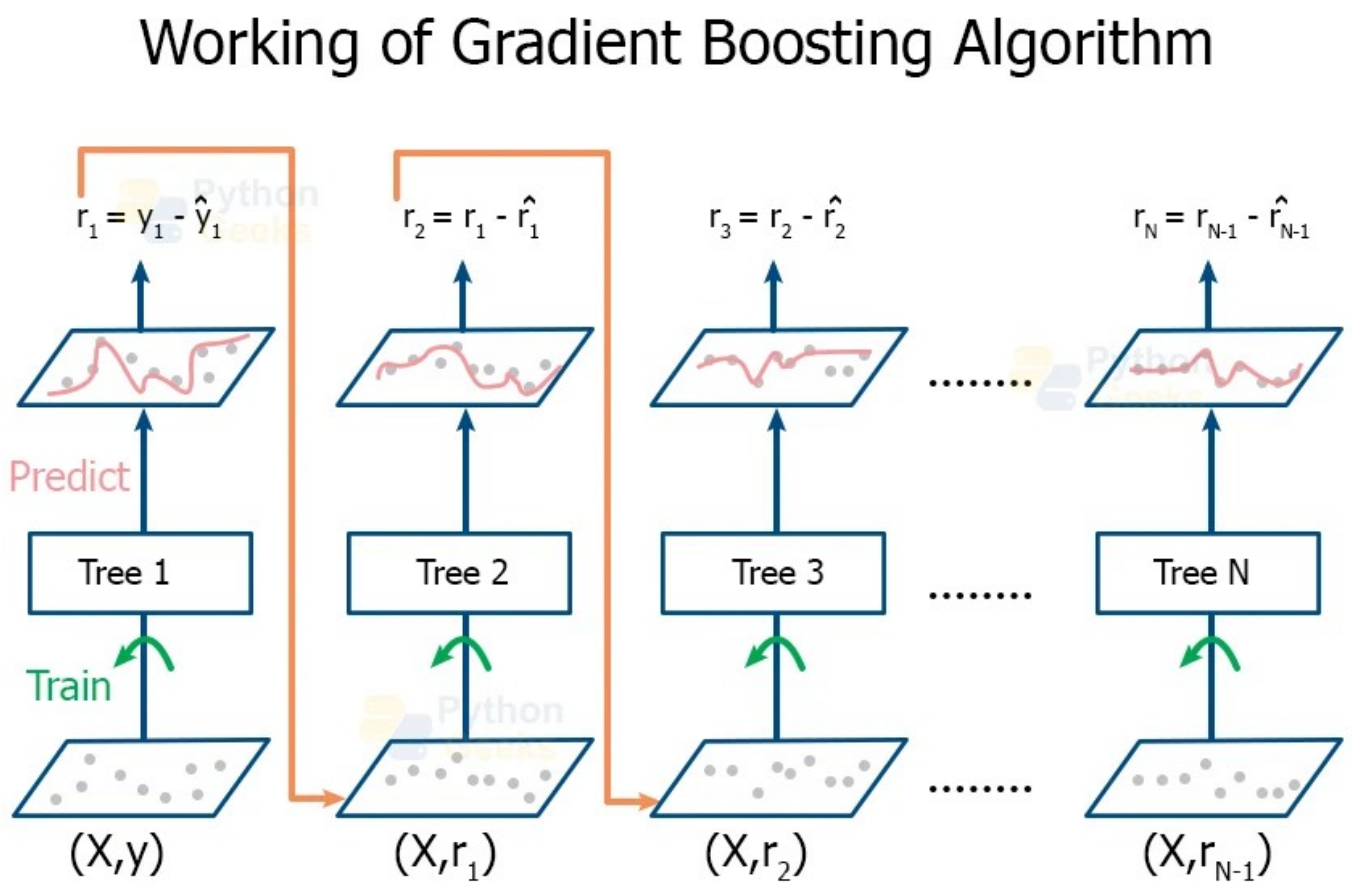
1. Initialize $f_0(x) = 0$
2. For $p \in \{1, \ldots, P^{boost}\}$
   (a) For $i \in \{1, \ldots, n\}$ compute

$$g_{i,p} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_{p-1}} \tag{11}$$

   (b) Fit a regression tree to $\{(x_i, g_{i,p})\}_{i=1}^{n}$, giving leaves l_{jp} for $j = 1, \ldots, J_p$
   (c) Compute optimal predictions for each leaf $j \in \{1, \ldots, J_p\}$:

$$\gamma_{jp} = \arg\min_{\gamma \in \mathbb{R}} \sum_{x_i \in l_{jp}} L(y_i, f_{p-1}(x_i) + \gamma) = \frac{1}{n_{l_{jp}}} \sum_{x_i \in l_{jp}} (y_i - f_{p-1}(x_i)) \tag{12}$$

   (d) Denote by $\tilde{f}_p(x) = \sum_{j=1}^{J_p} \mathbf{1}\{x \in l_{jp}\}\gamma_{jp}$ the predictions of the tree built in this fashion
   (e) Set $f_p(x) = f_{p-1}(x) + \eta \tilde{f}_p(x)$
3. Output $f(x) = f_{P^{boost}}(x)$

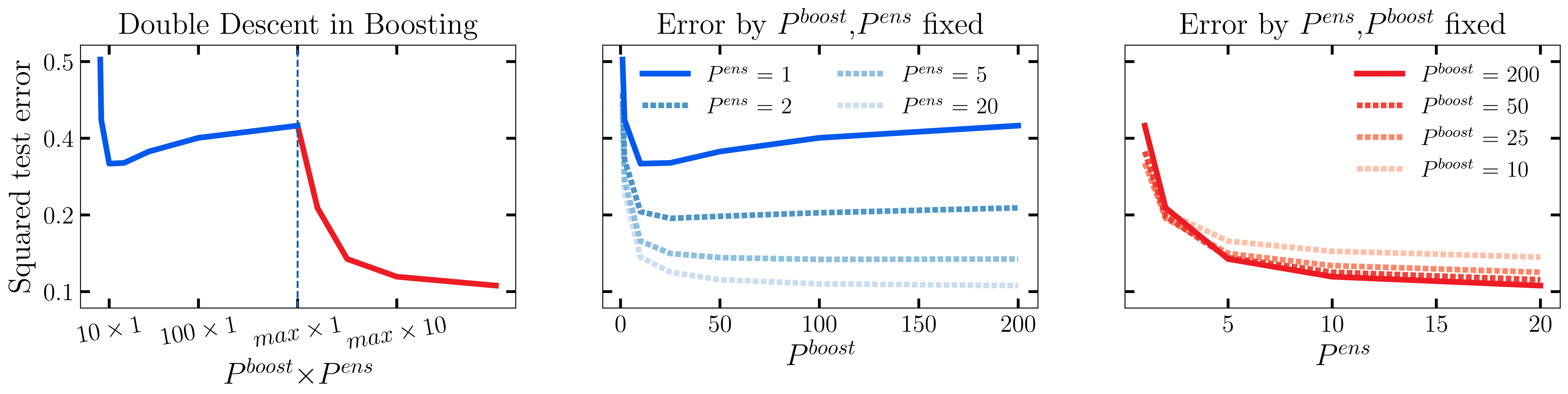


https://pythongeeks.org/gradient-boosting-algorithm-in-machine-learning/

- Boosting: sequentially training weak learners and constructing a strong (ensembled) model
- Recursively learning and accumulating “residuals” with weak learners
- A <u>gradient</u> of squared loss == (negative) residual! ➡️ use gradients for general loss.

Double descent in gradient boosting 💨

- P^{boost} : number of boosting rounds (number of weak learners to make a single final model)
- P^{ens} : number of independent final models for ensembling



Double descent in Linear regression 🎯

A review on Linear regression (1)

- Ordinary linear regression:
  - Inputs $\mathbf{X} \in \mathbb{R}^{n \times d}$ and labels $\mathbf{y} \in \mathbb{R}^n$ to learn $\boldsymbol{\beta} \in \mathbb{R}^d$ that fits $\mathbf{y} = \mathbf{X}\boldsymbol{\beta}$.
  - ⚠️ $P = d$ …!

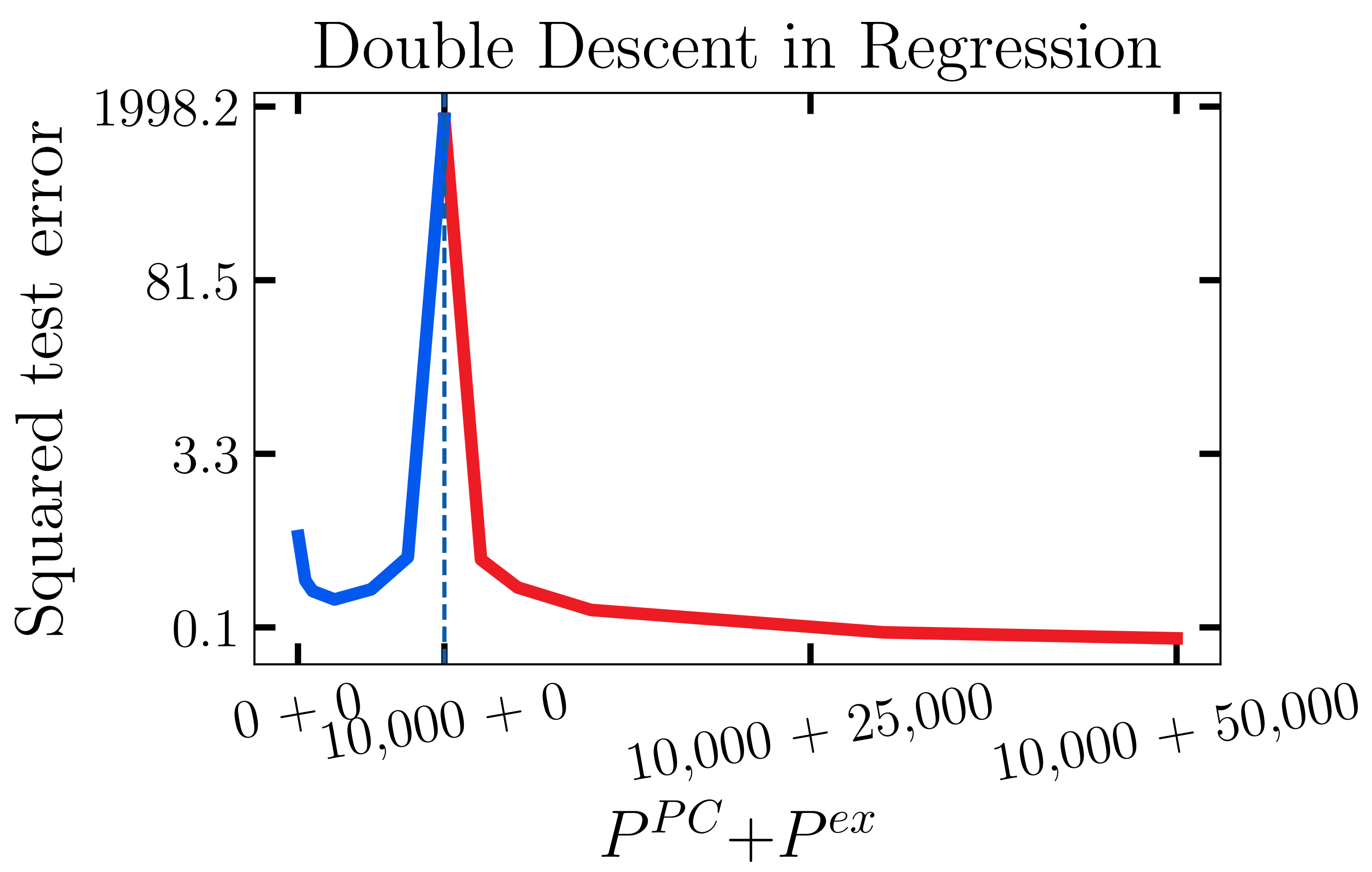
Double descent in Linear regression 🎯

A review on Linear regression (2)

- Linear regression with Random Fourier Features (RFF)
  - Given a feature matrix $\mathbf{\Phi} \in \mathbb{R}^{n \times P^{\phi}}$, learn $\boldsymbol{\beta} \in \mathbb{R}^{P^{\phi}}$ that fits $\mathbf{y} = \mathbf{\Phi}\boldsymbol{\beta}$.
  - The feature matrix $\mathbf{\Phi} = \left[\phi_j(\mathbf{x}_i)\right]_{i,j}$ is randomly generated as:
    - $\phi_j(\mathbf{x}) = \Re[\exp(\sqrt{-1}\mathbf{v}_j^{\top}\mathbf{x})]$ for all $j \in [P^{\phi}]$, where each $\mathbf{v}_j \overset{\text{iid}}{\sim} \mathcal{N}(\mathbf{0}, (1/5)\mathbf{I}_d)$.
- Still, there appears to be only one way to increase the number of parameters.

Double descent in Linear regression 🎯

A review on Linear regression (3)



- For $P^{\phi} \leq n$, there is a unique (least-square) solution $\hat{\boldsymbol{\beta}} = (\boldsymbol{\Phi}^{\top}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\top}\mathbf{y}$.
- For $P^{\phi} > n$, there are infinitely many solutions.
  - [BHMM19] rely on the minimum-ℓ_2-norm solution $\hat{\boldsymbol{\beta}} = \boldsymbol{\Phi}^{\top}(\boldsymbol{\Phi}\boldsymbol{\Phi}^{\top})^{-1}\mathbf{y}$.

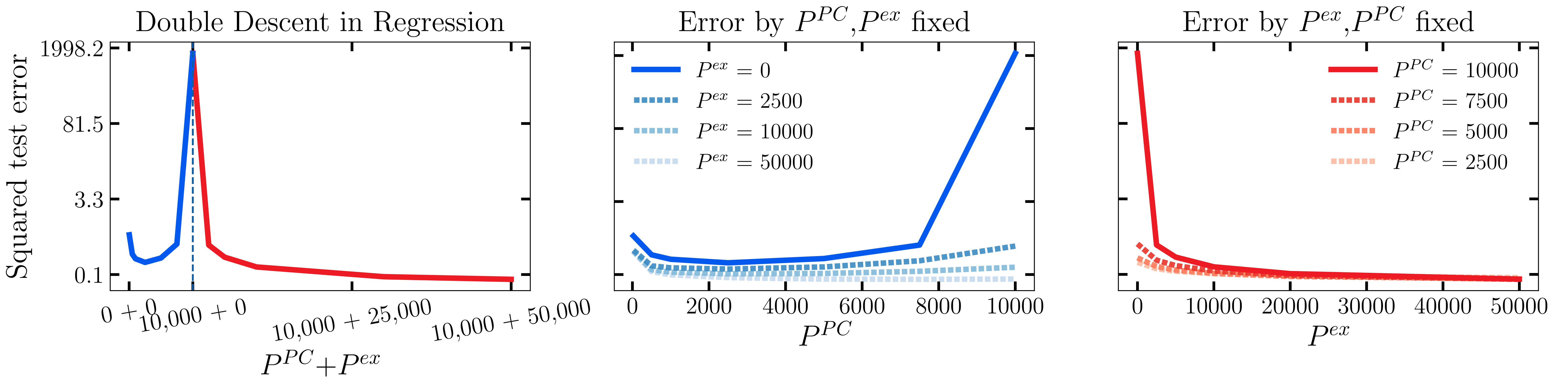
Double descent in Linear regression 🎯

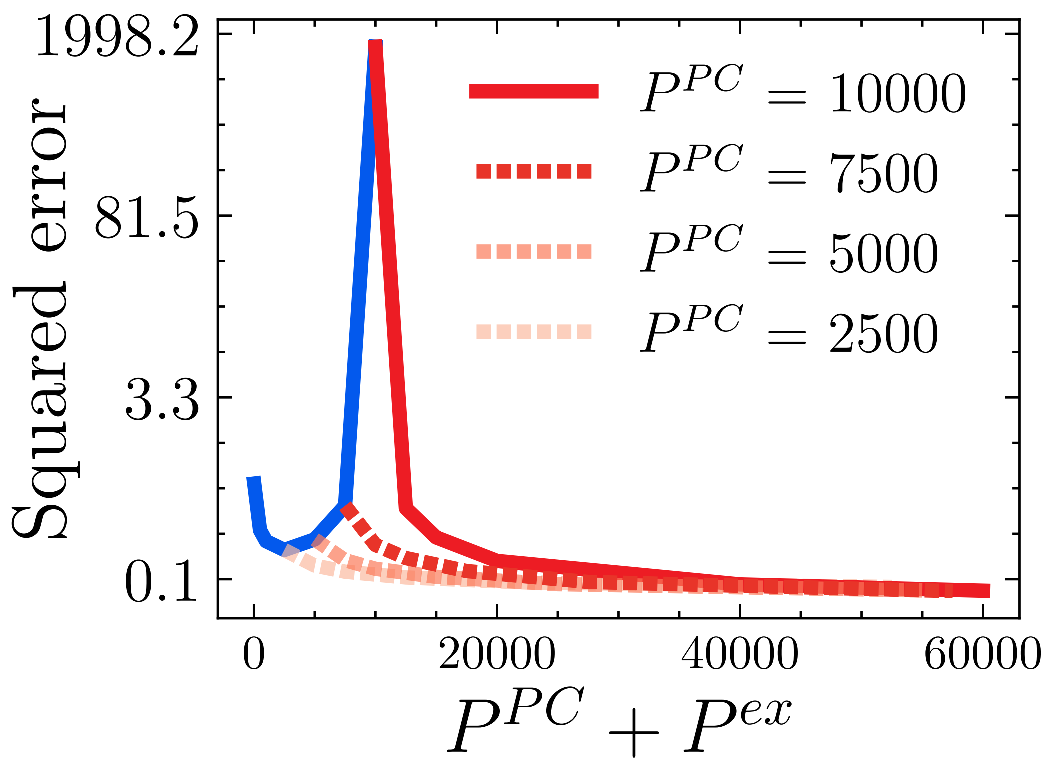
Connection between min-norm sol & dimensionality reduction

- The *true* parameter count does not increase in P^{ϕ} once $P^{\phi} > n$:
  - $\because \hat{\boldsymbol{\beta}} = \boldsymbol{\Phi}^{\top}(\boldsymbol{\Phi}\boldsymbol{\Phi}^{\top})^{-1}\mathbf{y}$ is a projection of $\mathbf{y}$ onto the row space of $\boldsymbol{\Phi}$ (whose rank is at most n).
- Finding a min-norm solution (when $P^{\phi} > n$) can be thought of as two steps:
  1. Unsupervised step: Dimensionality reduction of data into n-dimension
  2. Supervised step: Fitting n-dimensional regression coefficient
- Linear regression with RFF is thus a special case of Principal Component (PC) Regression.
  - Dimensionality reduction to $P^{PC} \leq \min\{P^{\phi}, n\}$ dimension while removing excess $P^{ex} = P^{\phi} - P^{PC}$ dimensions.
  - Then perform Ordinary Least Squares on the space with a reduced dimension.

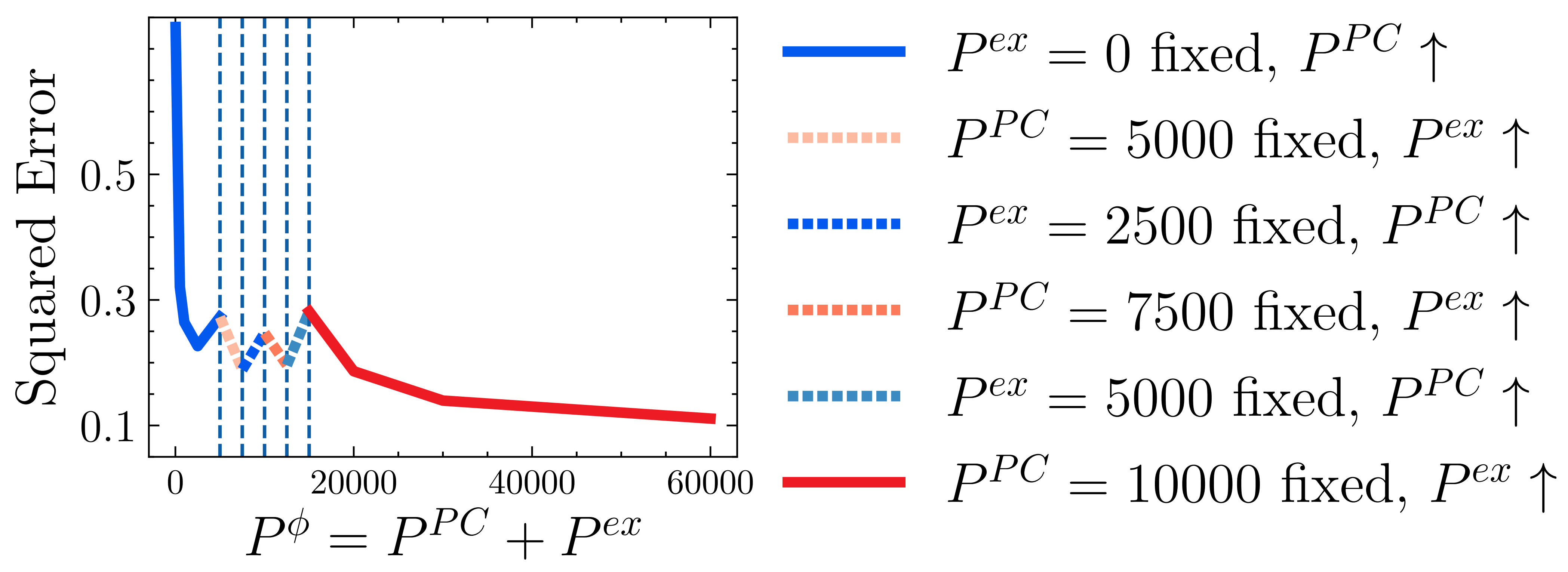
Double descent in Linear regression 🎯

Experiments





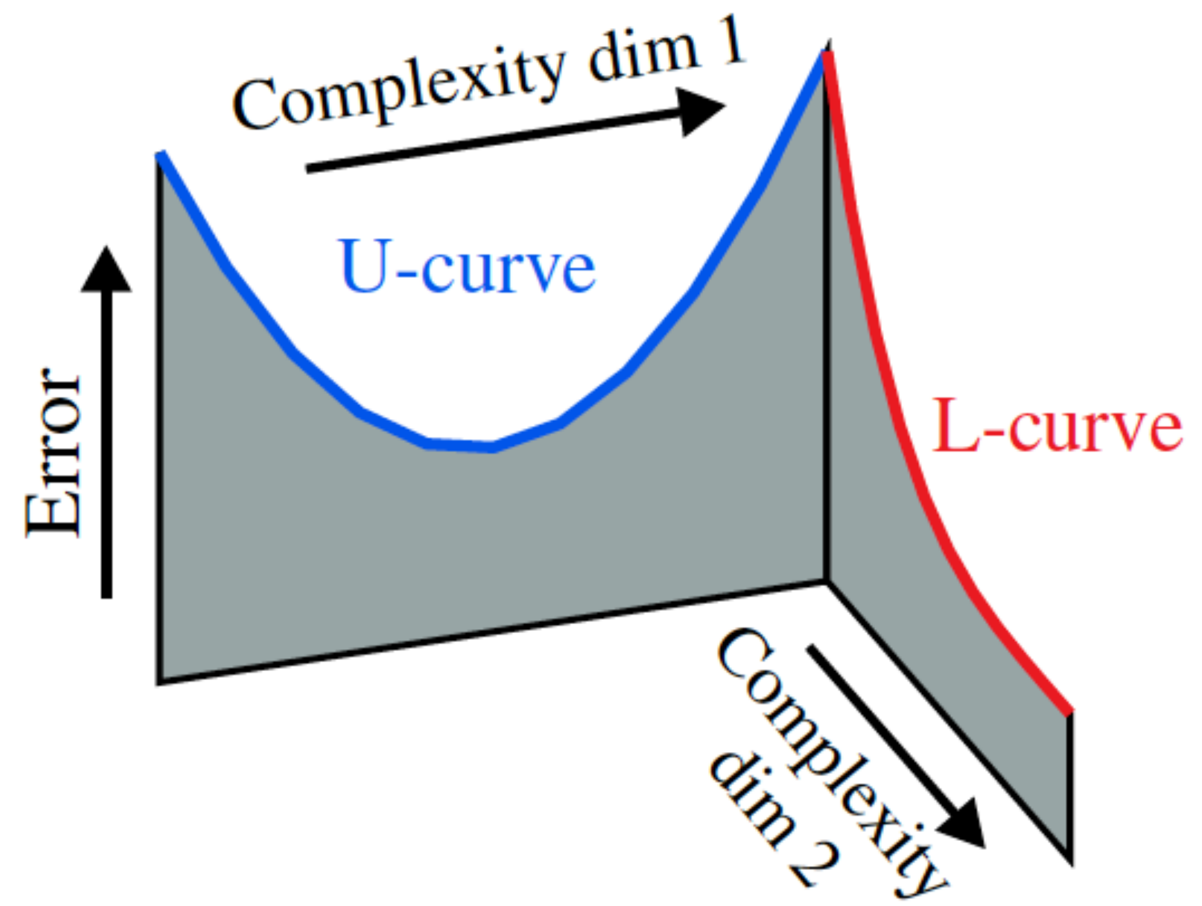
Moving the peak



Multiple descents

Part 1. Revisiting existing results [BHMM19]

Revisiting the evidence for double descent in non-deep ML models



- “There’s *more than one* complexity axis along which the param count grows.”
- “The location of the second descent is not tied to the interpolation threshold.”
  - …but tied to the transition of the complexity axis.

Part 2: Rethinking parameter counting

Through a classical statistics lens

- Redefine a measure of the effective number of parameters of a model: "generalized effective parameter measure" $p^0_{\hat{\mathbf{s}}}$
- Using $p^0_{\hat{\mathbf{s}}}$ to measure complexity, the double descent curves fold back into traditional U-shapes!

Smoothers

A unifying non-parametric statistical framework

- Let $\mathscr{D}^{train} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, $\mathbf{y}_{\text{train}} = [y_1, \ldots, y_n]^\top$
- The prediction of a **smoother** [HT90]: $\hat{f}(\tilde{\mathbf{x}}) = \hat{\mathbf{s}}(\tilde{\mathbf{x}})^\top \mathbf{y}_{\text{train}}$
    - $\hat{\mathbf{s}}(\tilde{\mathbf{x}}) \in \mathbb{R}^n$ is a smoother's weight
    - $\hat{s}^i(\tilde{\mathbf{x}}) = [\hat{\mathbf{s}}(\tilde{\mathbf{x}})]_i$ is a function of $\tilde{\mathbf{x}}$ and $(\mathbf{x}_i, y_i)$ (analogous to the concept of "kernel")
- Trees, boosting, and linear regressions: can be interpreted as smoothers. [CJvdS23]
- [CJvdS23] adapt the effective parameter definition for smoothers [HT90] and propose the <u>generalized effective parameter measure</u> (**Definition 1**): for an arbitrary set of inputs $\{\tilde{\mathbf{x}}_j\}_{j \in \mathscr{I}_{\text{test}}}$,

$$p_{\hat{\mathbf{s}}}^0 = \frac{n}{|\mathscr{I}_{\text{test}}|} \sum_{j \in \mathscr{I}_{\text{test}}} \|\hat{\mathbf{s}}(\tilde{\mathbf{x}}_j)\|^2$$

Smoothers

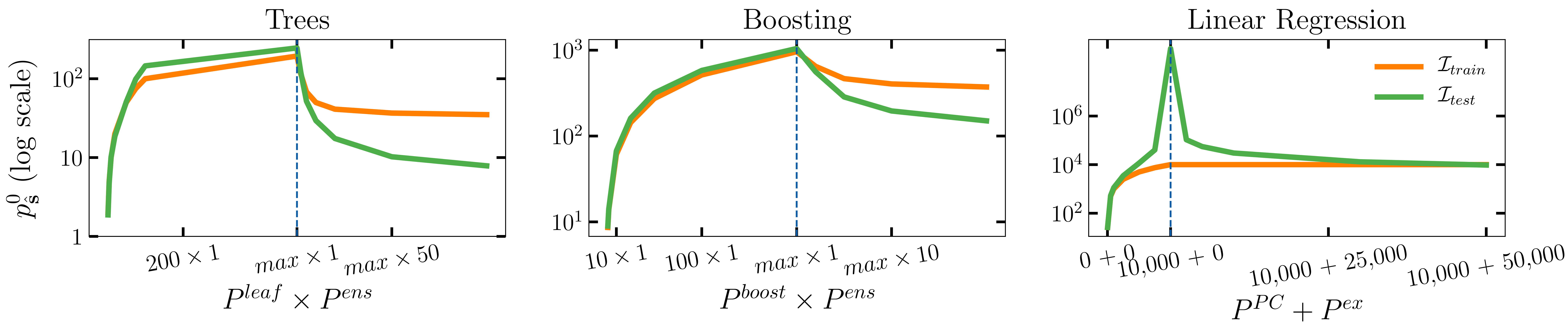
A brief motivation on generalized effective parameter measure

- An effective number of parameters is originally defined just for the training set:

$$p_e = \sum_{i \in \mathcal{I}_{\text{train}}} \|\hat{\mathbf{s}}(\mathbf{x}_i)\|^2$$

- [CJvdS23] adapt this definition to an arbitrary set of inputs indexed by $\mathcal{I}_{\text{test}}$
- The scale of the original p_e depends on n: the recalibration factor $\dfrac{n}{|\mathcal{I}_{\text{test}}|}$ is introduced.

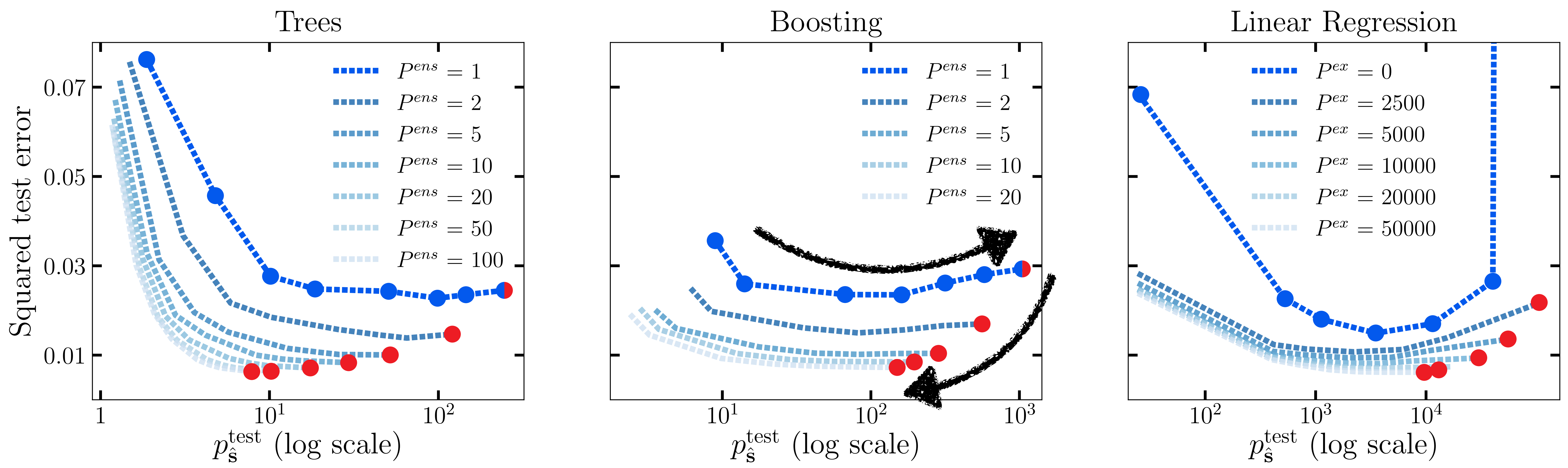
Raw v.s. effective number of parameters?



- The effective number of parameters measured on the test set does NOT increase as the raw parameter count increases.

Back to U: “U-turn” on Double Descent curve

- Putting $p^0_{\hat{\mathbf{s}}}$ (instead of raw parameter count) to the x-axis in [BHMM19]’s setup:



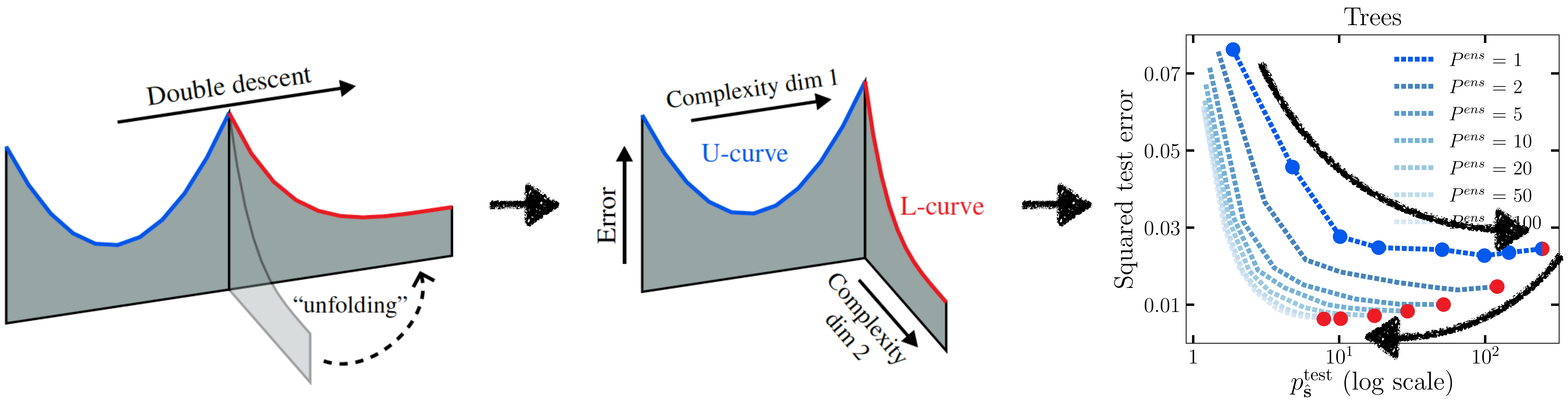
- blue points 🔵 : first axis $(P^{leaf}, P^{boost}, P^{PC})$

- red points 🔴 : second axis $(P^{ens}, P^{ens}, P^{ex})$

Part 2: Rethinking parameter counting

Through a classical statistics lens

- Redefine a measure of the effective number of parameters of a model: "generalized effective parameter measure" $p^0_{\hat{\mathbf{s}}}$
- Using $p^0_{\hat{\mathbf{s}}}$ to measure complexity, the double descent curves fold back into traditional U-shapes!

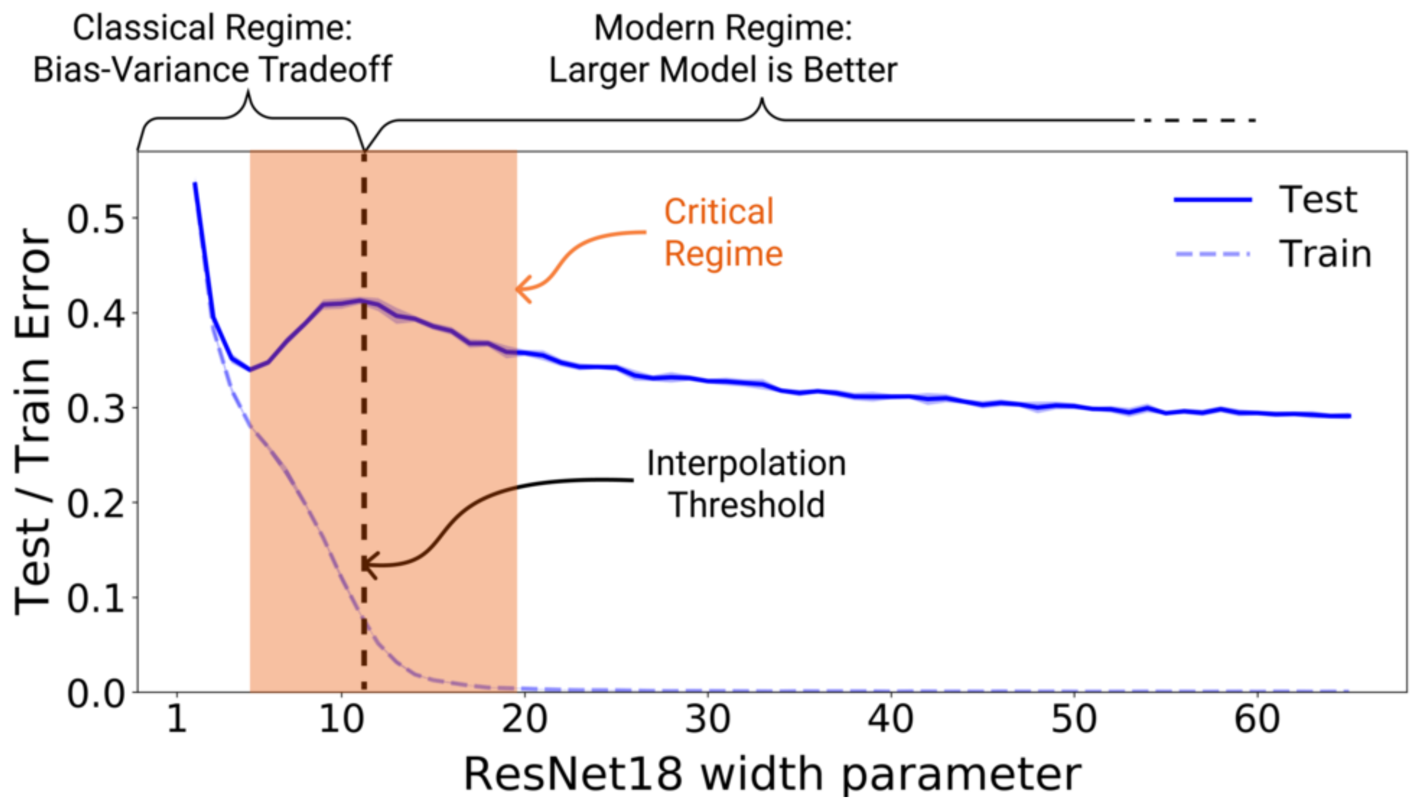


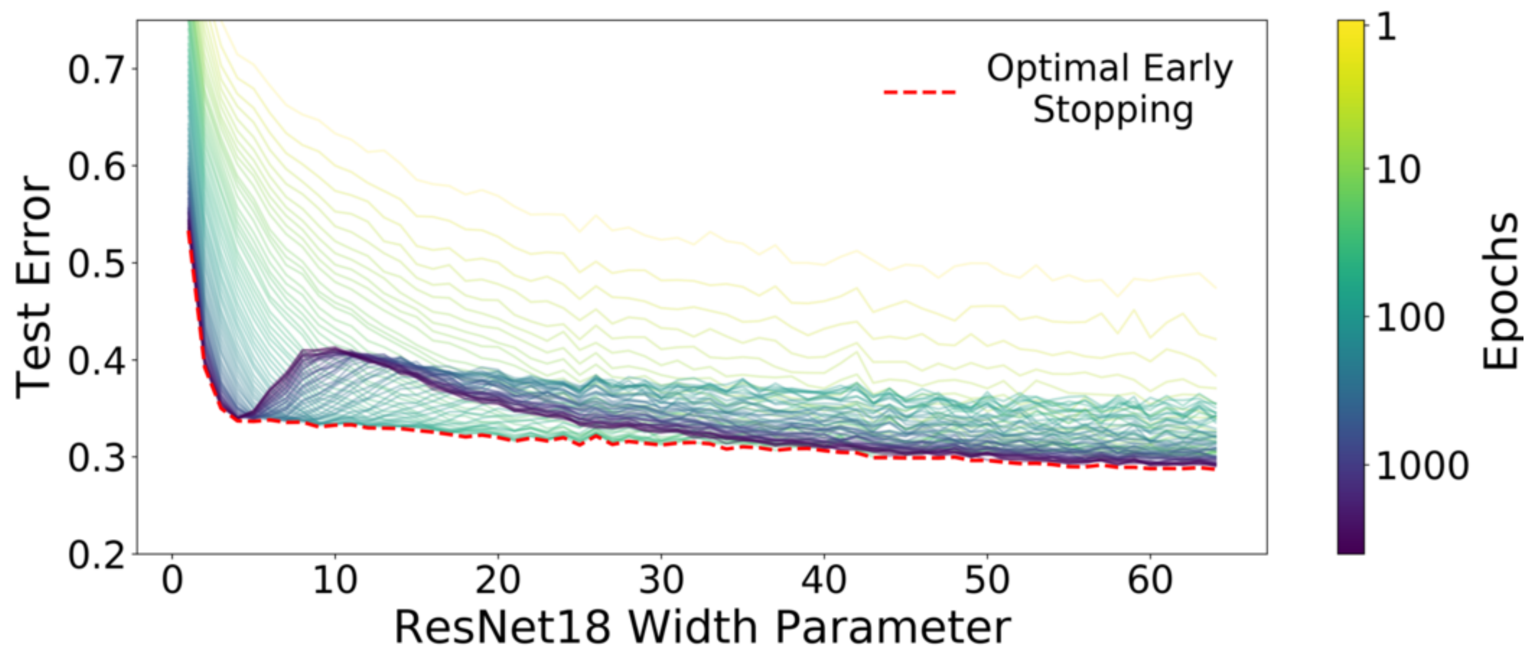
Double Descent

U-turn

Discussion

- A resolution of the tension between non-deep double descent and classical statistical intuition on U-shaped curve.
- Still, “Deep Double Descent [NKB+21]” is out of the scope.
  - In deep learning, Double descent occurs in terms of #param & #epochs





References

- [BHMM19] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.
- [CJvdS23] Alicia Curth, Alan Jeffares, and Mihaela van der Schaar. A U-turn on double descent: Rethinking parameter counting in statistical learning. In Thirty-seventh Conference on Neural Information Processing Systems, 2023.
- [HT90] Trevor Hastie and Robert Tibshirani. Generalized additive models. *Monographs on statistics and applied probability. Chapman & Hall*, 43:335, 1990.
- [NKB+21] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. Journal of Statistical Mechanics: Theory and Experiment, 2021(12):124003, 2021.