25. Topics in ML Theory

EEC454 Introduction to Machine Learning Systems

2023 Fall, Jaeho Lee

Hutchinson, “Has artificial intelligence become alchemy?” Science, 2018

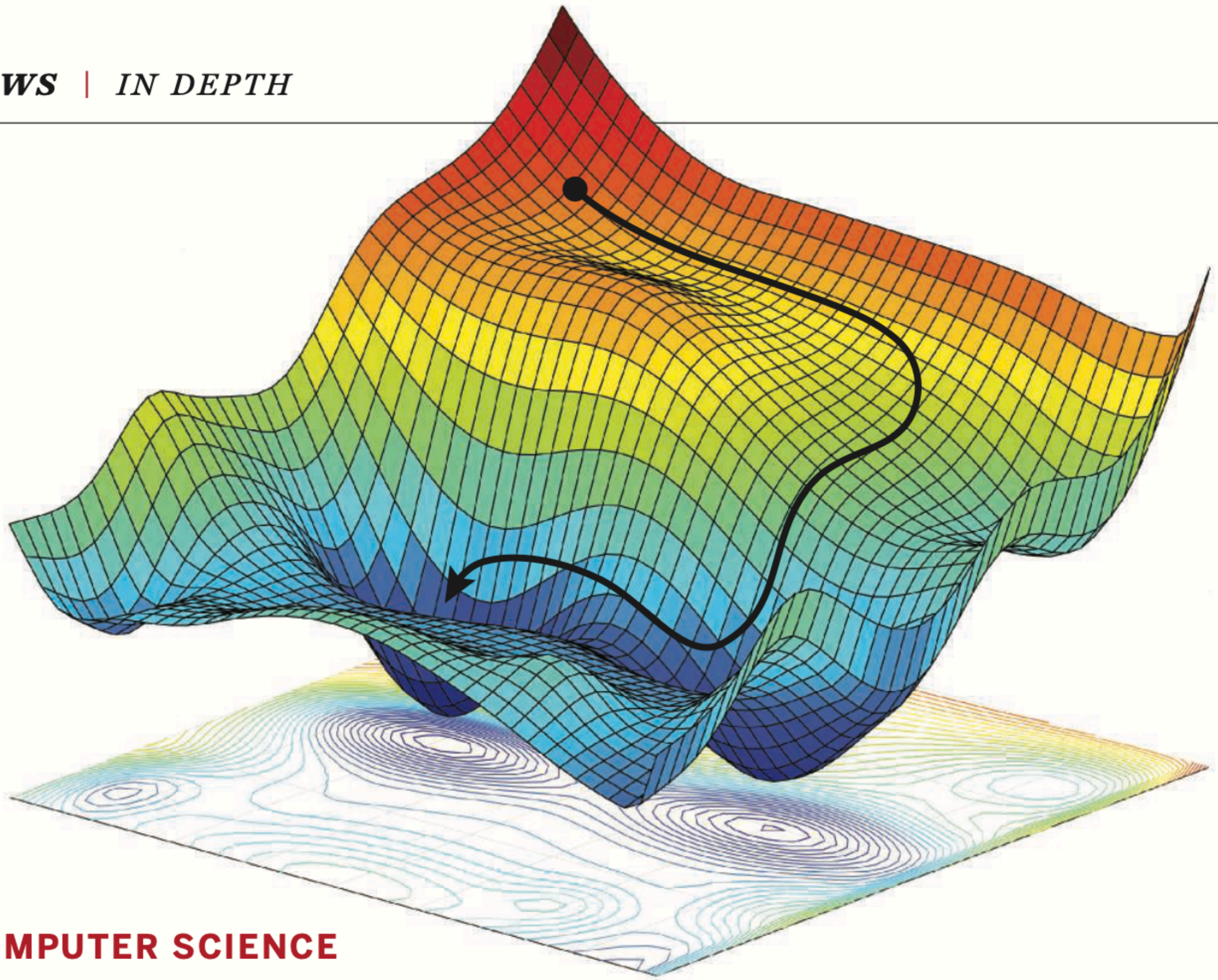
By now...

You might have noticed that
ML involves much engineering.

Evidences.

- Your own experience.
- NeurIPS 2017 test-of-time award titled *“ML has become alchemy”* by Ali Rahimi
- The science magazine article ->

NEWS | *IN DEPTH*



COMPUTER SCIENCE

Has artificial intelligence become alchemy?

Machine learning needs more rigor, scientists argue

Point of Criticism

The (continued) lack of theoretical understanding on DL.

2017

UNDERSTANDING DEEP LEARNING REQUIRES RETHINKING GENERALIZATION

Chiyuan Zhang*
Massachusetts Institute of Technology
chiyuan@mit.edu

Samy Bengio
Google Brain
bengio@google.com

Moritz Hardt
Google Brain
mrtz@google.com

Benjamin Recht†
University of California, Berkeley
brecht@berkeley.edu

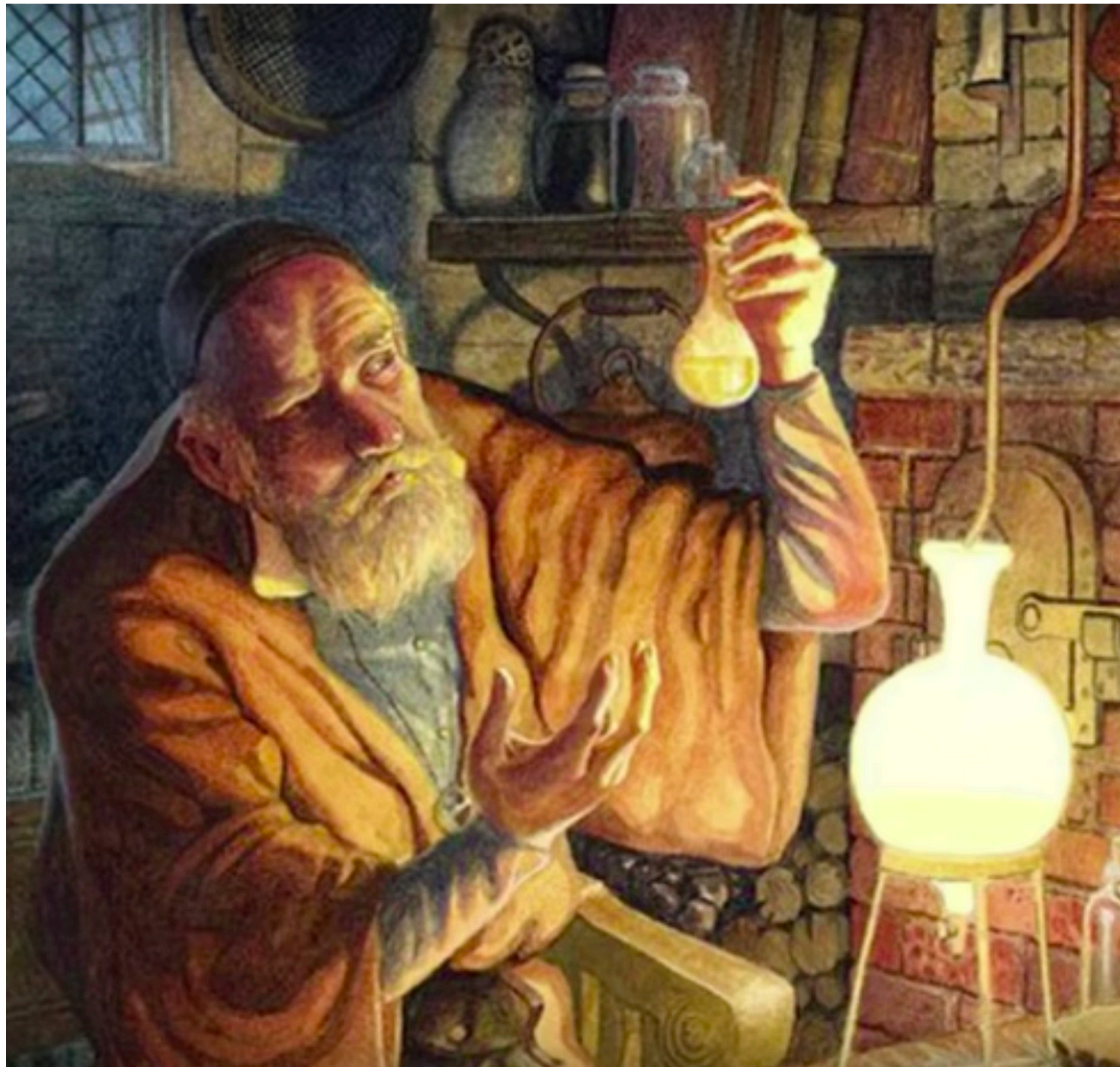
Oriol Vinyals
Google DeepMind
vinyals@google.com

2021

DOI:10.1145/3446776

Understanding Deep Learning (Still) Requires Rethinking Generalization

By **Chiyuan Zhang**, **Samy Bengio**, **Moritz Hardt**, **Benjamin Recht**, and **Oriol Vinyals**



Today

Still, many theoreticians are on a quest to

"mathematically formalize how deep learning works."

We take a glimpse on many topics of ***Machine Learning Theory***.

Basic Framework

Framework

A machine learning task can be described by three things:

- The dataset $D = \{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$
- The hypothesis space $\mathscr{F} = \{f_\theta : \theta \in \Theta\}$
- The loss function $\ell(f, \mathbf{z})$

Goal. Find a nice parameter $\hat{\theta}$ from D, such that

$$\mathbb{E}[\ell(f_{\hat{\theta}}, \mathbf{z})] \approx \min_{\theta \in \Theta} \underbrace{\mathbb{E}[\ell(f_\theta, \mathbf{z})]}_{=:L(\theta)}$$

Algorithm

ML algorithms are ***empirical risk minimization***, i.e., approximately solves

$$\underbrace{\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \ell(f_\theta, \mathbf{z}_i)}_{=:\hat{L}(\theta)}$$

Reason. If we have many data, we have

$$L(\theta) \approx L(\hat{\theta}) \qquad \text{for any } \theta \in \Theta$$

Decomposing the "test risk"

We are interested in characterizing the ***test risk***, of the learned $\hat{\theta}$, i.e.,

$$L(\hat{\theta})$$

This can be broken down as:

$$\boxed{L(\hat{\theta}) - \min_{\theta \in \Theta} L(\theta)} + \boxed{\min_{\theta \in \Theta} L(\theta)}$$

- : Excess risk
- : Minimum error one could get from the hypothesis space. (***approximation***)

Decomposing the “excess risk”

The excess risk can be decomposed as

$$
\begin{aligned}
&L(\hat{\theta}) - L(\theta^*) \\
&= L(\hat{\theta}) - \hat{L}(\hat{\theta}) + \hat{L}(\hat{\theta}) - \hat{L}(\theta^*) + \hat{L}(\theta^*) - L(\theta^*)
\end{aligned}
$$

- : How similar test risk is to training risk.
  (***Generalization***)

Decomposing the “excess risk”

The excess risk can be decomposed as

$$L(\hat{\theta}) - L(\theta^*)$$

$$= L(\hat{\theta}) - \hat{L}(\hat{\theta}) + \hat{L}(\hat{\theta}) - \hat{L}(\theta^*) + \hat{L}(\theta^*) - L(\theta^*)$$

- : How similar test risk is to training risk. (*Generalization*)

The yellow term can be further decomposed as:

$$\hat{L}(\hat{\theta}) - \hat{L}(\theta^*) = \left(\hat{L}(\hat{\theta}) - \min_{\theta \in \Theta} \hat{L}(\theta)\right) + \min_{\theta \in \Theta} \hat{L}(\theta) - \hat{L}(\theta^*)$$

How well $\hat{\theta}$ solves ERM (***Optimization***)

<= zero, always

Three elements of Learning theory

From this perspective, ***learning theory*** is primarily about developing mathematical tools for three objects:

- **Approximation.** $\min_{\theta \in \Theta} L(\theta)$
- **Generalization.** $\hat{L}(\hat{\theta}) - L(\hat{\theta})$
- **Optimization.** $\hat{L}(\hat{\theta}) - \min_{\theta \in \Theta} \hat{L}(\theta)$

Three elements of Learning theory

From this perspective, *learning theory* is primarily about developing mathematical tools for three objects:

- **Approximation.** $\min_{\theta \in \Theta} L(\theta)$
- **Generalization.** $\hat{L}(\hat{\theta}) - L(\hat{\theta})$
- **Optimization.** $\hat{L}(\hat{\theta}) - \min_{\theta \in \Theta} \hat{L}(\theta)$

If Θ is very big. we expect **approximation** 👍
generalization 👎
optimization 🤔

Three elements of Learning theory

From this perspective, *learning theory* is primarily about developing mathematical tools for three objects:

- **Approximation.** $\min_{\theta \in \Theta} L(\theta)$
- **Generalization.** $\hat{L}(\hat{\theta}) - L(\hat{\theta})$
- **Optimization.** $\hat{L}(\hat{\theta}) - \min_{\theta \in \Theta} \hat{L}(\theta)$

If Θ is very big. we expect **approximation** 👍
generalization 👎
optimization 🤔

Reality. All 👍 for DL!

Approximation

Approximation

Formal version.

For any ground-truth function $g(\mathbf{z})$, # e.g., human label

there exists a nice parameter $\theta \in \Theta$ such that

$$\mathbb{E}[\|f_\theta(\mathbf{z}) - g(\mathbf{z})\|^2] < \epsilon$$

(or alternatively, $\sup_{\mathbf{z}} \|f_\theta(\mathbf{z}) - g(\mathbf{z})\| < \epsilon$)

Universal Approximation Theorem

DL. Several old results state that
two-layer neural network can approximate any function.
(given sufficient width)

Universal approximation theorem — Let $C(X, \mathbb{R}^m)$ denote the set of continuous functions from a subset X of a Euclidean $\mathbb{R}^n$ space to a Euclidean space $\mathbb{R}^m$. Let $\sigma \in C(\mathbb{R}, \mathbb{R})$. Note that $(\sigma \circ x)_i = \sigma(x_i)$, so $\sigma \circ x$ denotes σ applied to each component of x.

Then σ is not polynomial if and only if for every $n \in \mathbb{N}$, $m \in \mathbb{N}$, compact $K \subseteq \mathbb{R}^n$, $f \in C(K, \mathbb{R}^m)$, $\varepsilon > 0$ there exist $k \in \mathbb{N}$, $A \in \mathbb{R}^{k \times n}$, $b \in \mathbb{R}^k$, $C \in \mathbb{R}^{m \times k}$ such that

$$\sup_{x \in K} \|f(x) - g(x)\| < \varepsilon$$

where $g(x) = C \cdot (\sigma \circ (A \cdot x + b))$

UAT (depth ver.)

Recent works show that one can prove similar results for ***thin networks***, given sufficient depths.

MINIMUM WIDTH FOR UNIVERSAL APPROXIMATION

Sejun Park† **Chulhee Yun**‡ **Jaeho Lee**†* **Jinwoo Shin**†*

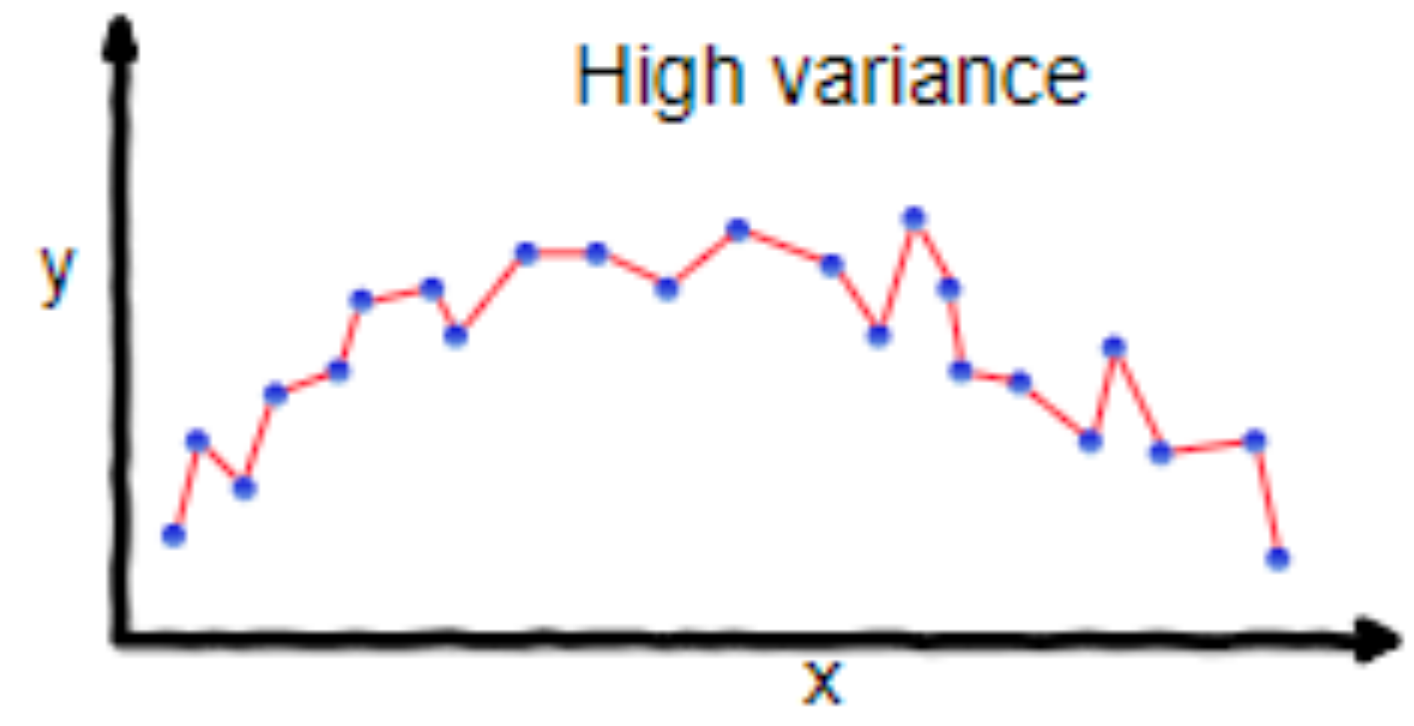
Reference	Function class	Activation ρ	Upper / lower bounds
Lu et al. (2017)	$L^1(\mathbb{R}^{d_x}, \mathbb{R})$	ReLU	$d_x + 1 \leq w_{\min} \leq d_x + 4$
	$L^1(\mathcal{K}, \mathbb{R})$	ReLU	$w_{\min} \geq d_x$
Hanin and Sellke (2017)	$C(\mathcal{K}, \mathbb{R}^{d_y})$	ReLU	$d_x + 1 \leq w_{\min} \leq d_x + d_y$
Johnson (2019)	$C(\mathcal{K}, \mathbb{R})$	uniformly conti.†	$w_{\min} \geq d_x + 1$
Kidger and Lyons (2020)	$C(\mathcal{K}, \mathbb{R}^{d_y})$	conti. nonpoly‡	$w_{\min} \leq d_x + d_y + 1$
	$C(\mathcal{K}, \mathbb{R}^{d_y})$	nonaffine poly	$w_{\min} \leq d_x + d_y + 2$
	$L^p(\mathbb{R}^{d_x}, \mathbb{R}^{d_y})$	ReLU	$w_{\min} \leq d_x + d_y + 1$
Ours (Theorem 1)	$L^p(\mathbb{R}^{d_x}, \mathbb{R}^{d_y})$	ReLU	$w_{\min} = \max\{d_x + 1, d_y\}$
Ours (Theorem 2)	$C([0,1], \mathbb{R}^2)$	ReLU	$w_{\min} = 3 > \max\{d_x + 1, d_y\}$
Ours (Theorem 3)	$C(\mathcal{K}, \mathbb{R}^{d_y})$	ReLU+Step	$w_{\min} = \max\{d_x + 1, d_y\}$
Ours (Theorem 4)	$L^p(\mathcal{K}, \mathbb{R}^{d_y})$	conti. nonpoly‡	$w_{\min} \leq \max\{d_x + 2, d_y + 1\}$

† requires that ρ is uniformly approximated by a sequence of one-to-one functions.
‡ requires that ρ is continuously differentiable at some z with $\rho'(z) \neq 0$.

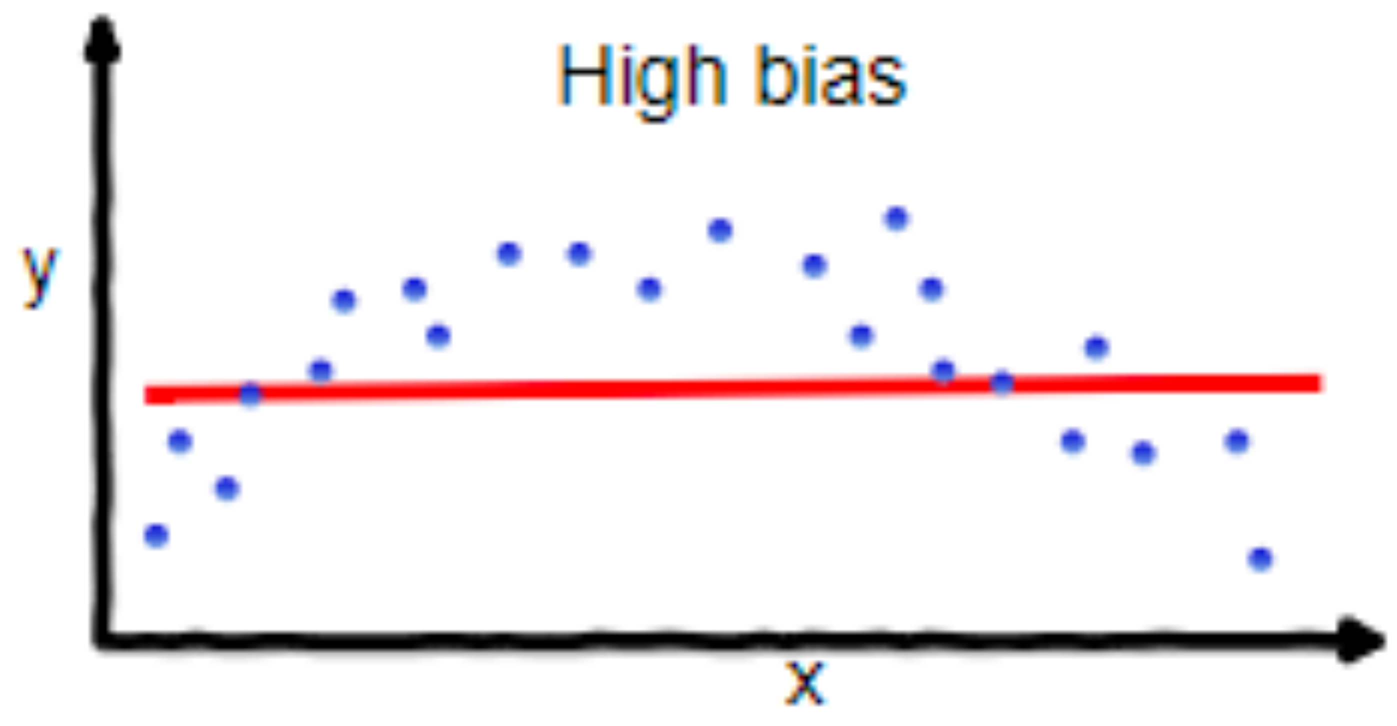
Generalization

Generalization

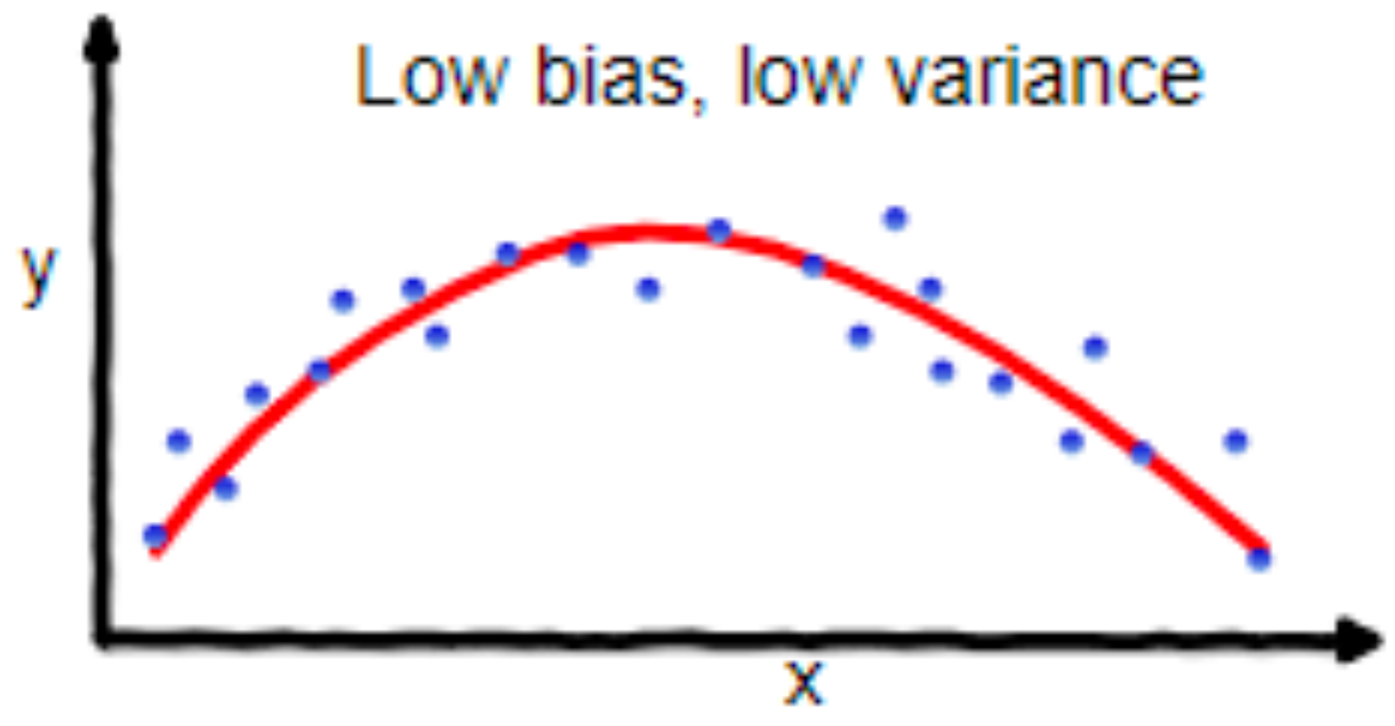
Classic idea. If there are too many parameters,
the learned function should overfit.



overfitting



underfitting

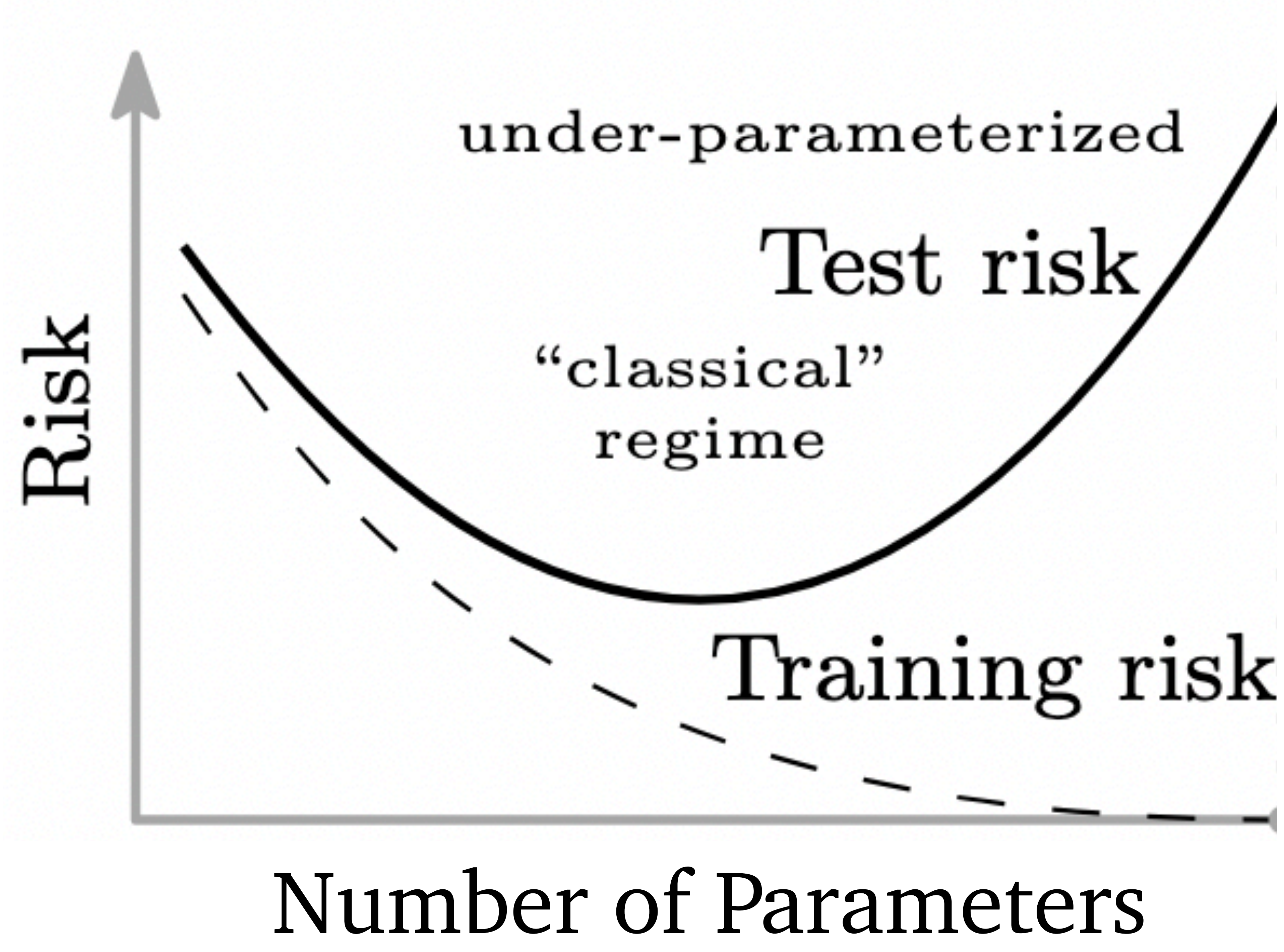


Good balance

Generalization

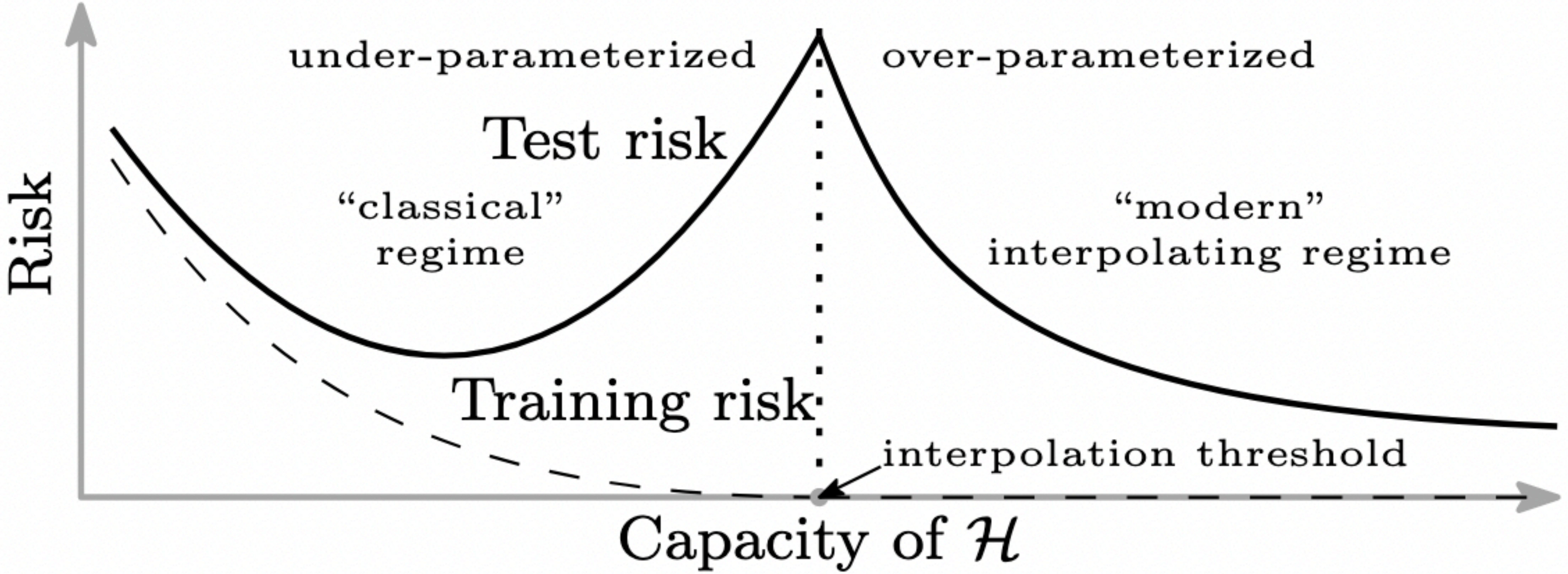
Classic results. With high probability, we have …

$$\sup_{\theta} |L(\theta) - \hat{L}(\theta)| \leq C \cdot \sqrt{\frac{\log|\Theta|}{n}}$$



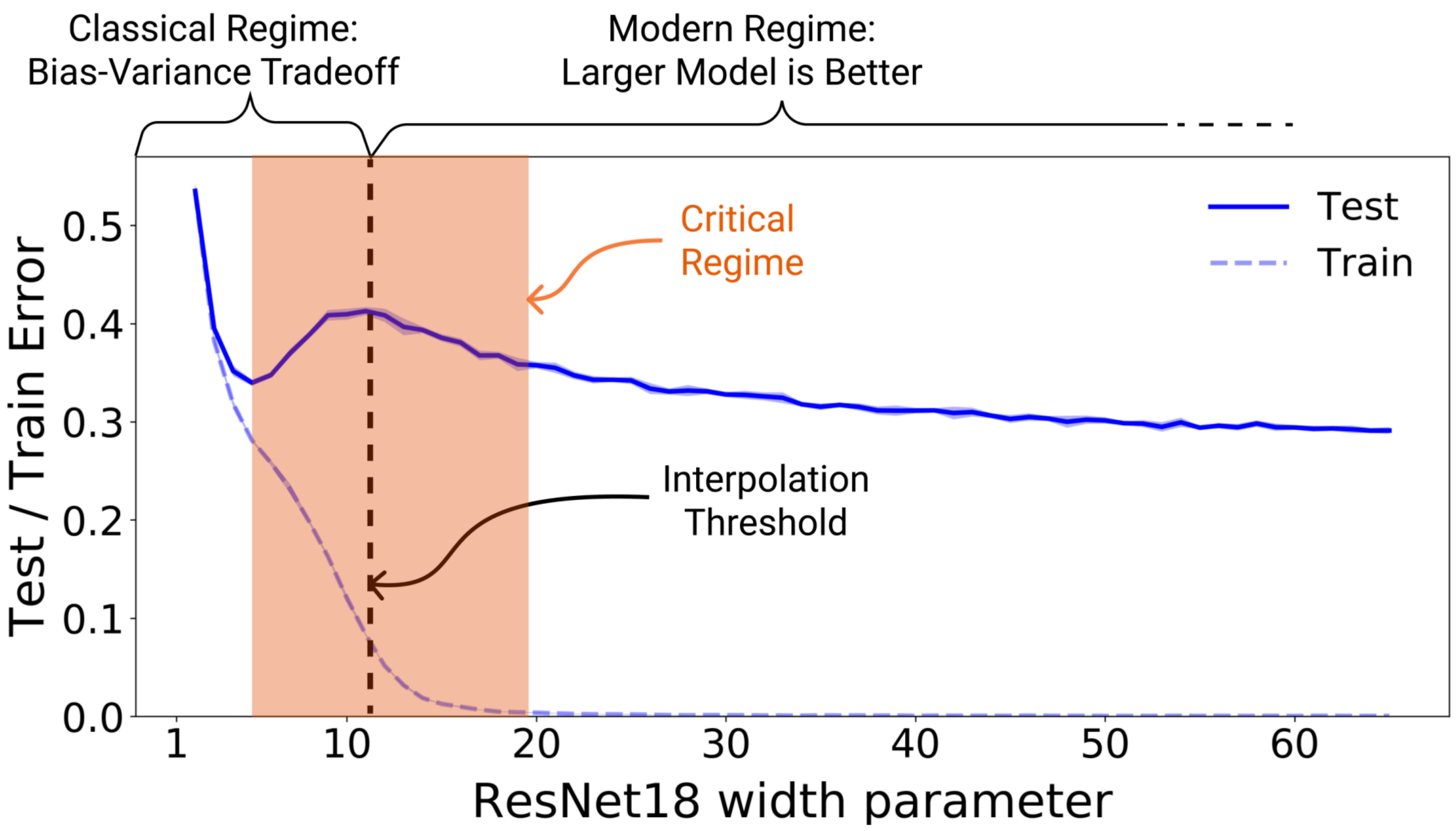
Generalization

DL. Generalization gets better with more parameters



Generalization

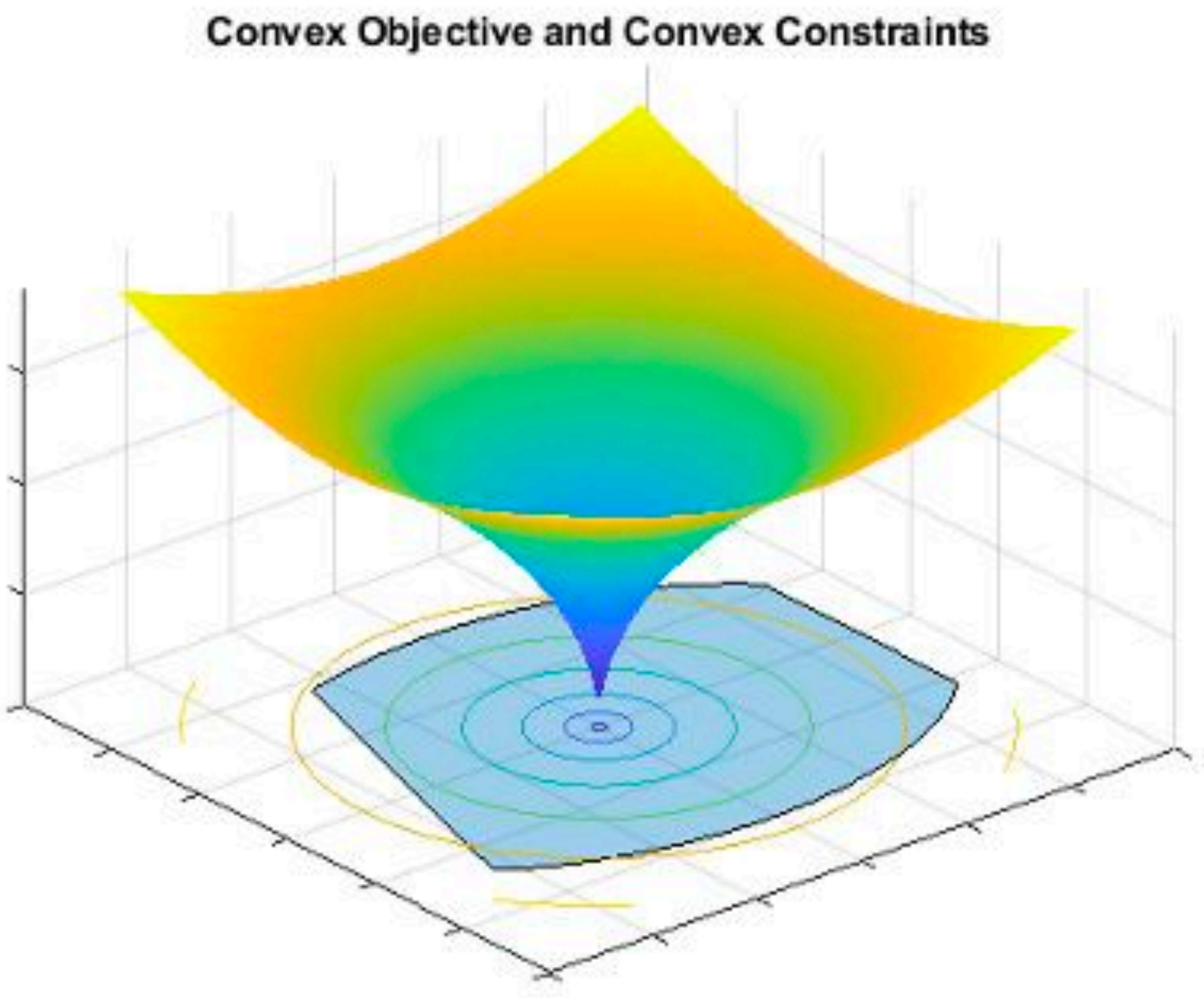
DL. Generalization gets better with more parameters?
(still not fully understood!)

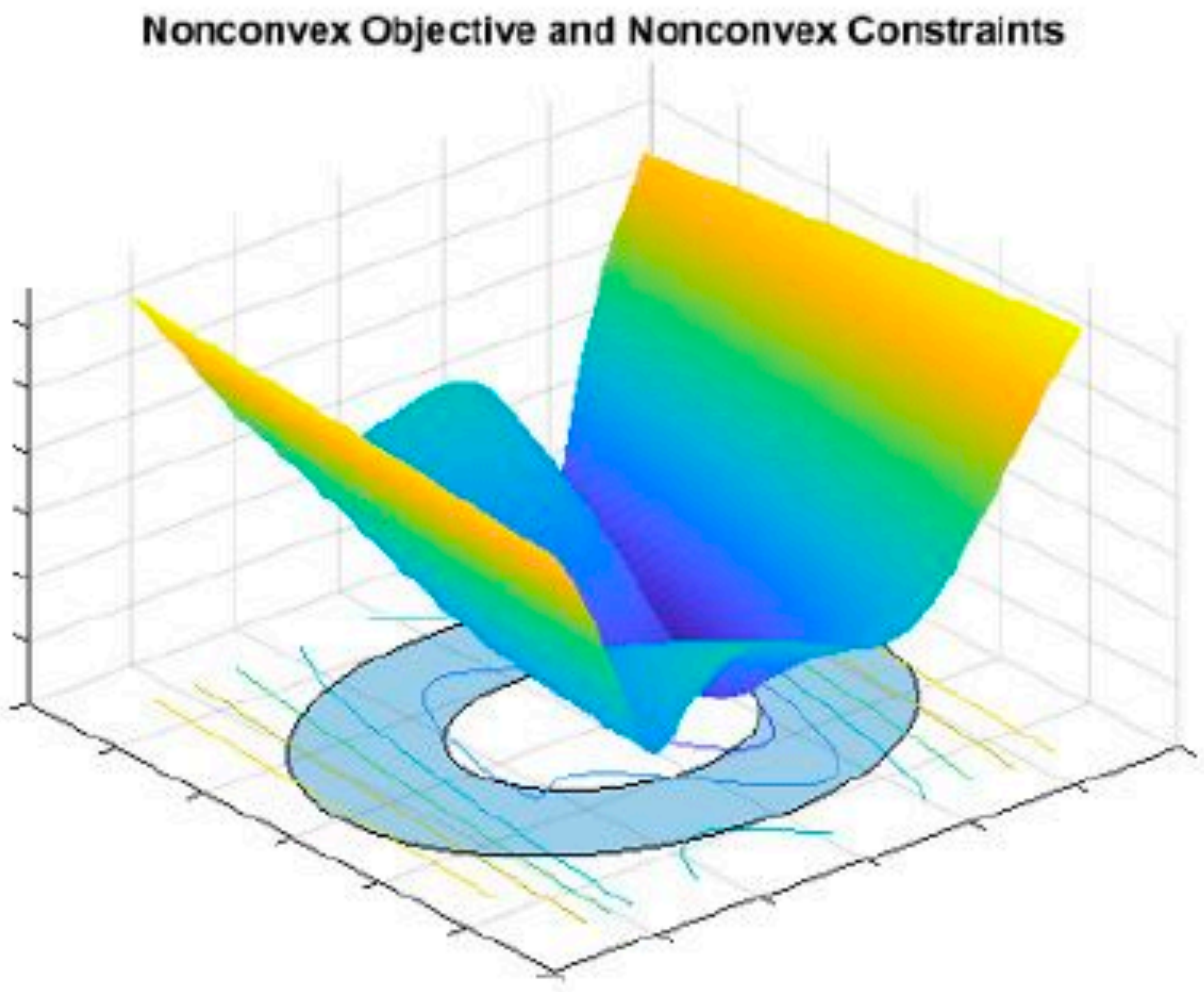


Optimization

Optimization

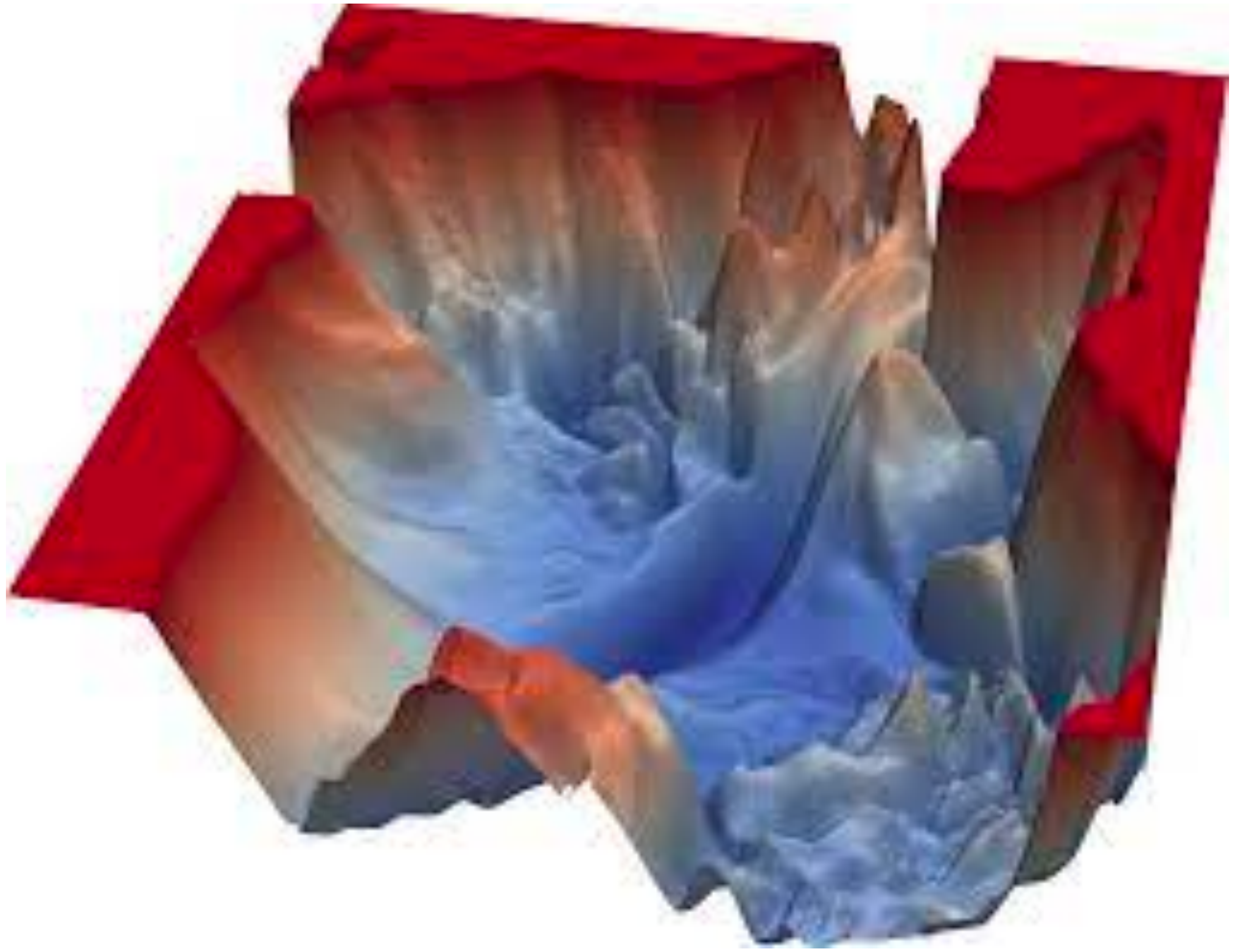
Classic idea. If convex, SGD converges well.
If nonconvex, SGD may not really converge.





Optimization

DL. Highly nonconvex, yet converges well
(especially for very big models)



Remarks

Concluding Remarks

- ML is still full of mysteries.
  - Especially because you need to handle ***data***
    (highly random and difficult to characterize; no Gaussian works!)
  - Still needs some ***alchemy***.
    - Part annoying, part fun.

- Waiting for new challengers to unravel the mystery…

Cheers