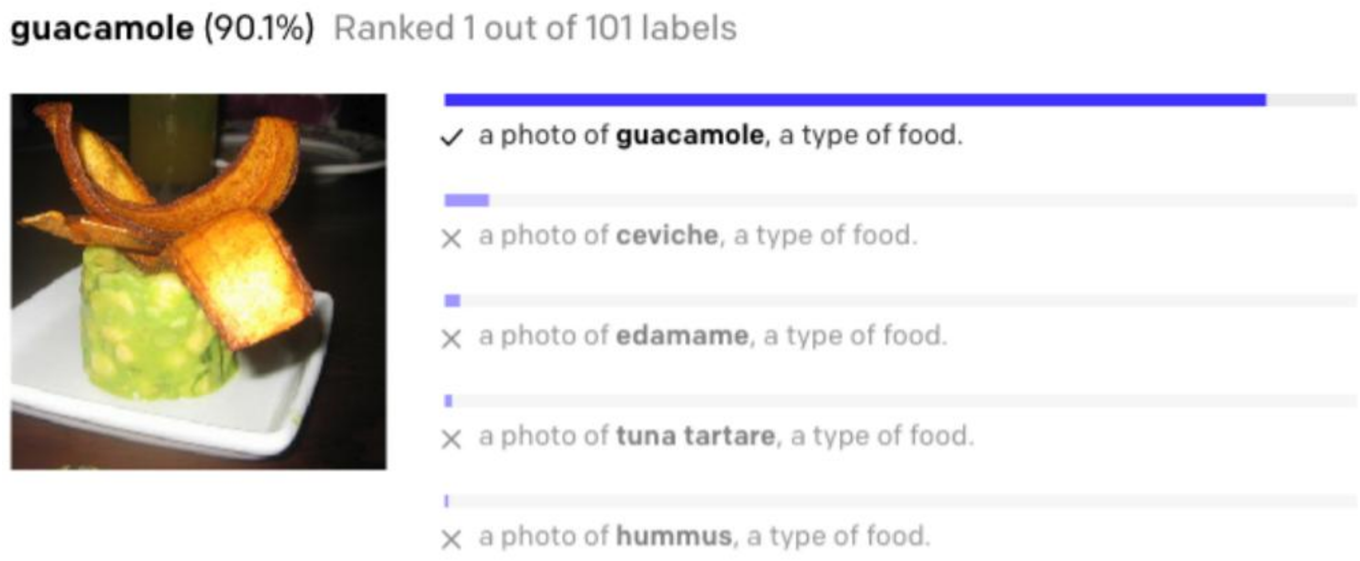
Towards A Statistical Understanding of Neural Network Classifiers

Tianyang Hu
Huawei Noah's Ark Lab

2023.07.07 @ Center for Statistical Science, Tsinghua University

Deep Learning

Deep learning has great empirical success.



Multi-Modal CLIP doing classification

OpenAI DALLE-2

User What is funny about this image? Describe it panel by panel.

Source: https://www.reddit.com/r/hmmm/comments/ubab5v/hmmm/

GPT-4 The image shows a package for a "Lightning Cable" adapter with three panels.

Panel 1: A smartphone with a VGA connector (a large, blue, 15-pin connector typically used for computer monitors) plugged into its charging port.

Panel 2: The package for the "Lightning Cable" adapter with a picture of a VGA connector on it.

Panel 3: A close-up of the VGA connector with a small Lightning connector (used for charging iPhones and other Apple devices) at the end.

The humor in this image comes from the absurdity of plugging a large, outdated VGA connector into a small, modern smartphone charging port.

OpenAI GPT-4 technical report

Machine Learning in a nutshell

Figure from https://xkcd.com/1838/

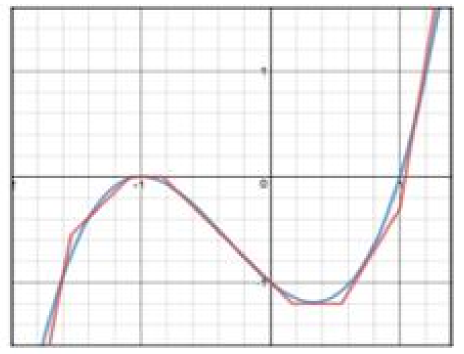
But theoretical understanding and guidance are still lacking.

Deep Learning Theory

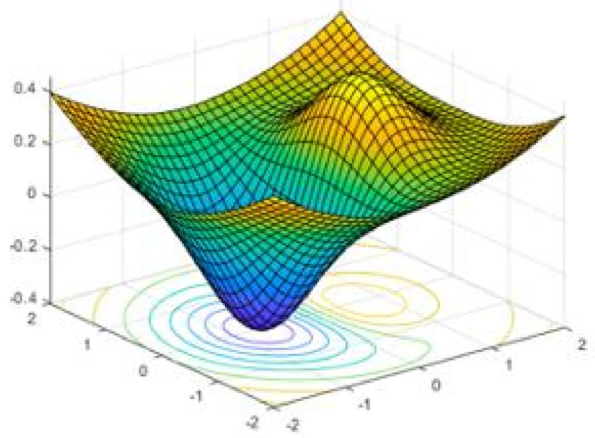
Why models built with neural networks can handle large-scale, high dimensional data extremely well?

How Deep Models work?

Expressibility

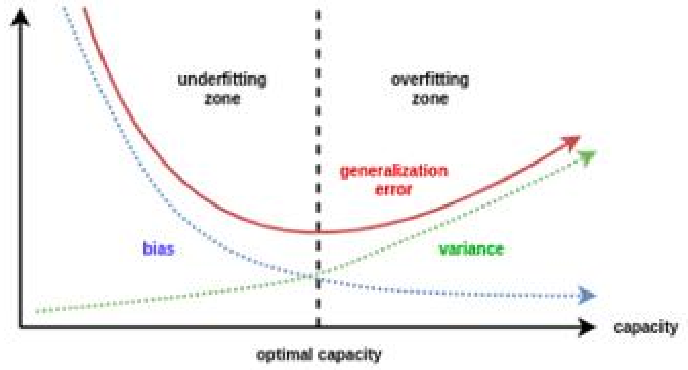


Universal approximation theorem
Deep neural network vs shallow neural network



Optimization

Gradient vanishing / exploding
GD can get stuck at saddle point.
Whether / how GD / SGD finds global optima?



Generalization

Generalization error bound based on Rademacher complexity
Does overparametrization hurt generalization?

Is this the whole story? Does Statistics play a big role in it?

Deep Learning Theory – A Statistics Perspective

For a certain task:

- What is the **estimation problem** and what are the most appropriate **ground truth assumptions**?
- What is the best we can do (**optimal sample complexity**)? Does the **curse of dimensionality** occur?
- Can DNNs achieve the best performance (**statistical optimality**)? Is there **algorithmic guarantees**?

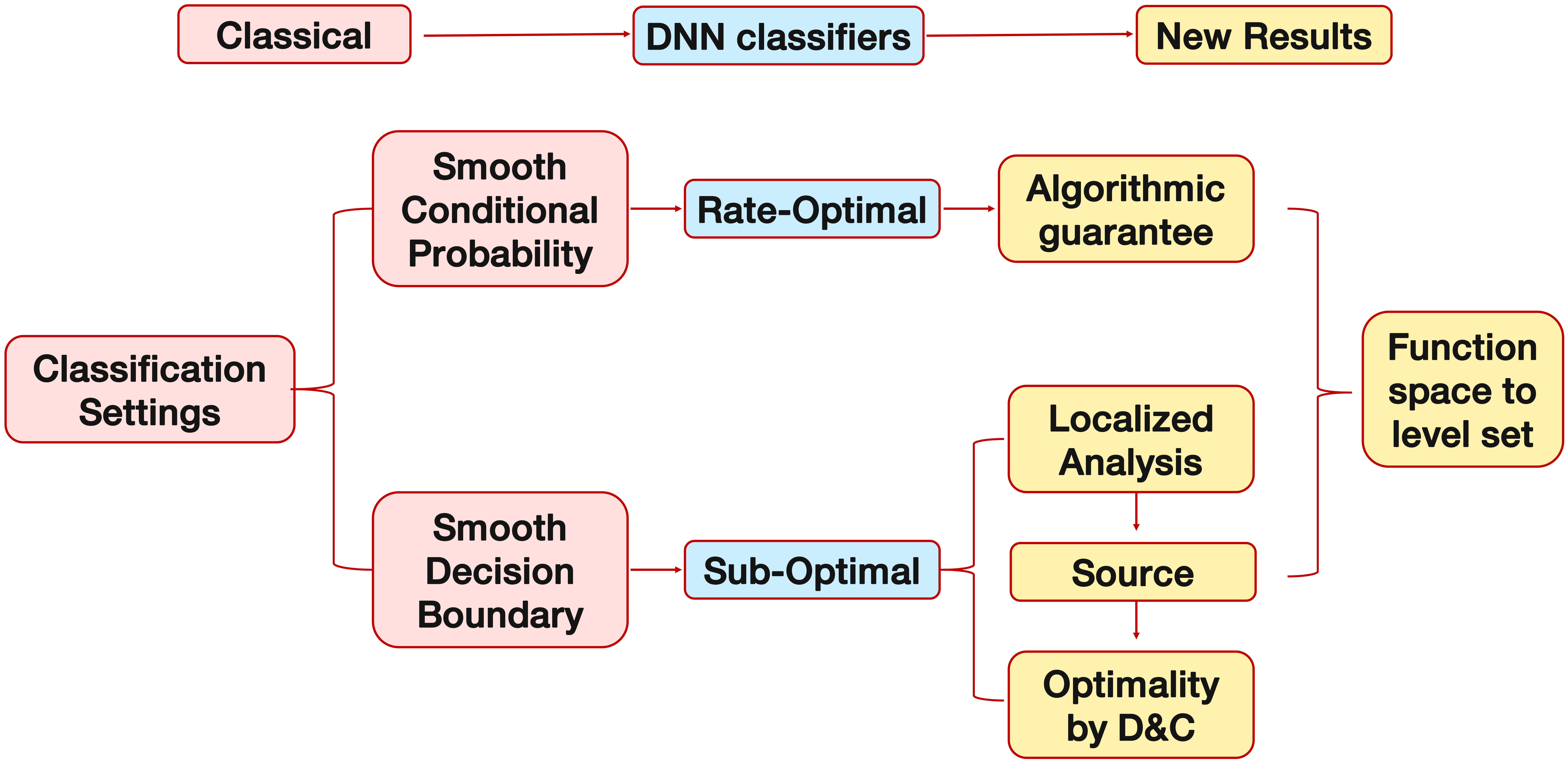
Why is Statistical Optimality Important?

- It can produce **sharp** characterization of the estimation method.
- It offers **fair comparison** between different models.
- It **complements** the other research areas revolving DNNs.

It's **underexplored** in the current mainstream research areas

Viewing DNN as a estimation tool, can they achieve **statistical optimal** rates in typical tasks, specifically **classification**?

Outline



- *Understanding Square Loss in Training Overparameterized Neural Network Classifiers*, **NeurIPS 2022 Spotlight**, with Wenjia Wang, Jun Wang, Zhenguo Li
- *Minimax Optimal Deep Neural Network Classifiers Under Smooth Decision Boundary*, arXiv, with Ruiqi Liu, Zuofeng Shang, Guang Cheng
- *Exact Count of Boundary Pieces of ReLU Classifiers:Towards the Proper Complexity Measure for Classification*, **UAI 2023**, with Pawel Piwek, Adam Klukowski

Binary Classification – Basic Settings

Let $x \in \mathbb{R}^d$ and $y \in \{-1, 1\}$ be the labels. Assume equal class probabilities and $x|y = 1 \sim p(x)$, $x|y = -1 \sim q(x)$. Observe data $\{(x_1, y_1) \cdots, (x_n, y_n)\}$.

- The optimal classifier $C^* := \mathrm{argmin}_{C \in \mathcal{C}} \mathbb{E}[1\{C(x) \neq y\}]$ can be written as the sign of $(p - q)$.
- Empirical minimization of 0-1 loss with is NP-hard, so surrogate loss is usually used in practice.
- DNN classifiers $\widehat{f} : \mathbb{R}^d \to \mathbb{R}$ can be trained using surrogate loss ϕ (hinge loss $\phi(z) = \max\{0, 1 - z\}$, cross entropy, etc.) by

$$\min_{f \in \mathcal{F}} \sum_{i=1}^{n} \phi(y_i f(x_i)).$$

Conditional Probability:

$$\eta(x) = P(y = 1|x) = \frac{p(x)}{p(x) + q(x)}$$

Decision Boundary:

$$\{x|\, \eta(x) = \frac{1}{2}\} \ or \ \{x|\, p(x) = q(x)\}$$

Optimal Classifier:

$$\{\eta(x) > \tfrac{1}{2}\} \ \textbf{or} \ \{2\eta(x) - 1 > 0\}$$

Excess Risk:

$$\mathcal{E}(\widehat{f}, C^*) = R(\widehat{f}) - R(C^*),$$

where $R(C)$ denotes the expected 0-1 risk $\mathbb{E}[1\{C(x) \neq y\}]$.

Binary Classification – Basic Settings

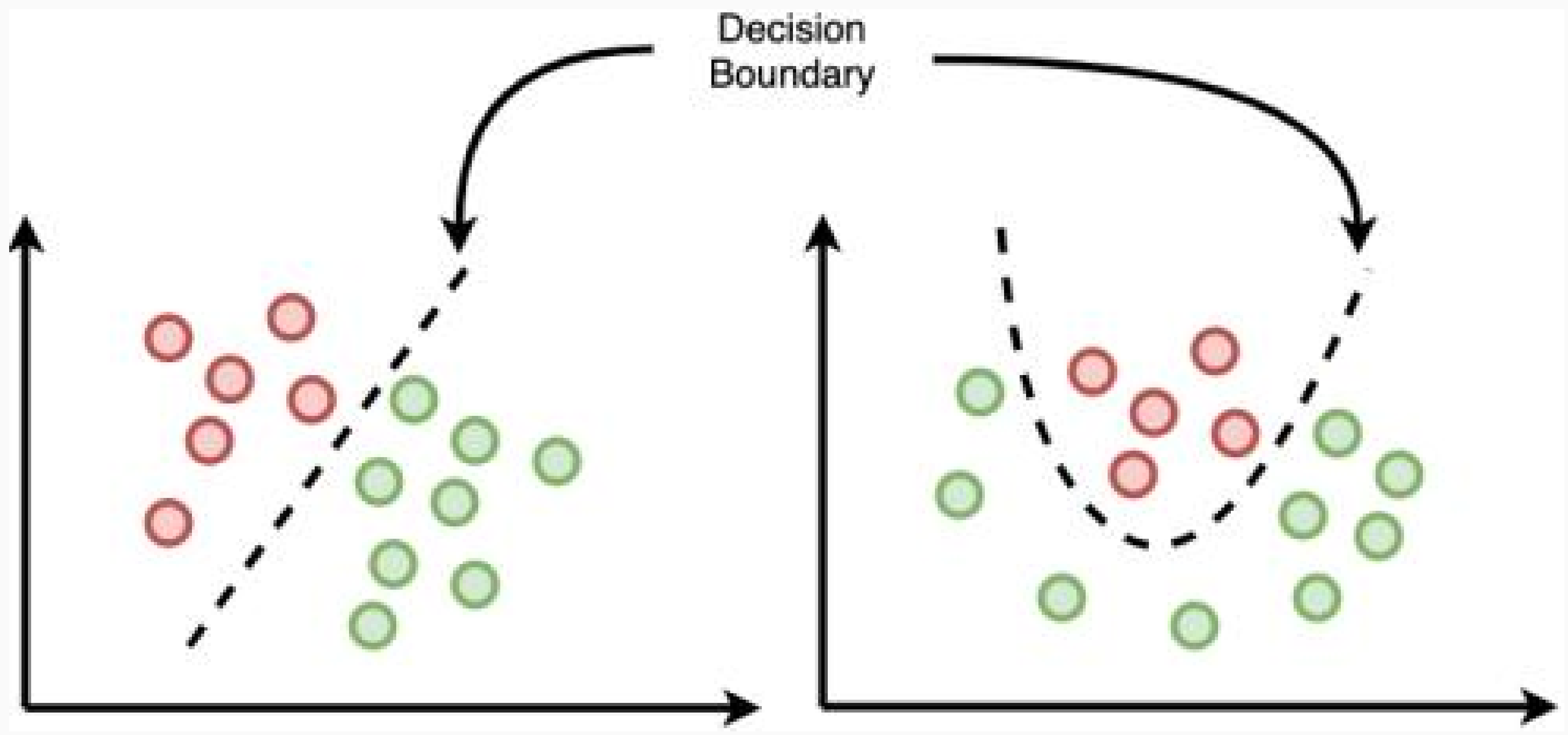
The performance of $\widehat{f}$ is measured by its **excess risk**

$$\mathcal{E}(\widehat{f}, C^*) = R(\widehat{f}) - R(C^*),$$

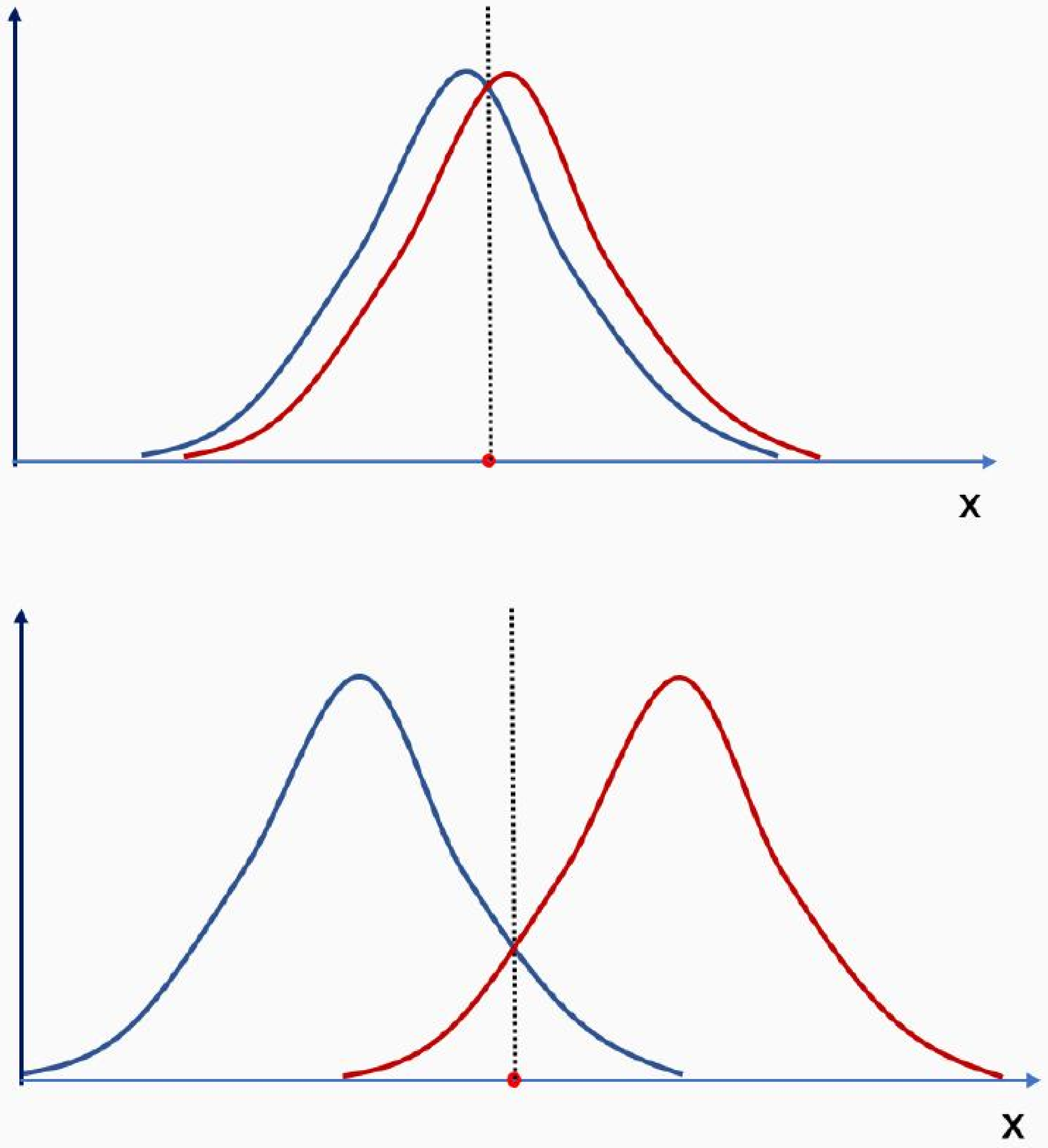
where $R(C)$ denotes the expected 0-1 risk $\mathbb{E}\left[1\{C(\mathbf{x}) \neq y\}\right]$.

There are two key factors governing the rate of convergence of the excess risk:

Complexity: covering number, smoothness, etc.



Separation: margin condition, Tsybakov's noise condition, etc.



Binary Classification – Complexity Assumptions

Complexity of the classification problem: Conditional probability vs. Decision Boundary

ASSUMPTION (CAR). The regression function η belongs to the class Σ of functions on $\mathbf{R}^d$ such that

$$\mathcal{H}(\varepsilon, \Sigma, L_p) \leq A_* \varepsilon^{-\rho} \qquad \forall \varepsilon > 0,$$

with some constants $\rho > 0$, $A_* > 0$. Here $\mathcal{H}(\varepsilon, \Sigma, L_p)$ denotes the ε-entropy of the set Σ w.r.t. an L_p norm with some $1 \leq p \leq \infty$.

ASSUMPTION (CAD). The decision set G^* belongs to a class $\mathcal{G}$ of subsets of $\mathbf{R}^d$ such that

$$\mathcal{H}(\varepsilon, \mathcal{G}, d_\triangle) \leq A_* \varepsilon^{-\rho} \qquad \forall \varepsilon > 0,$$

with some constants $\rho > 0$, $A_* > 0$. Here $\mathcal{H}(\varepsilon, \mathcal{G}, d_\triangle)$ denotes the ε-entropy of the class $\mathcal{G}$ w.r.t. the measure of symmetric difference pseudo-distance between sets defined by $d_\triangle(G, G') = P_X(G \triangle G')$ for two measurable subsets G and G' in $\mathbf{R}^d$.

Audibert & Tsybakov, Fast Learning Rate for Plug-in Classifiers, Annals of Statistics 2007

Binary Classification – Separation Assumptions

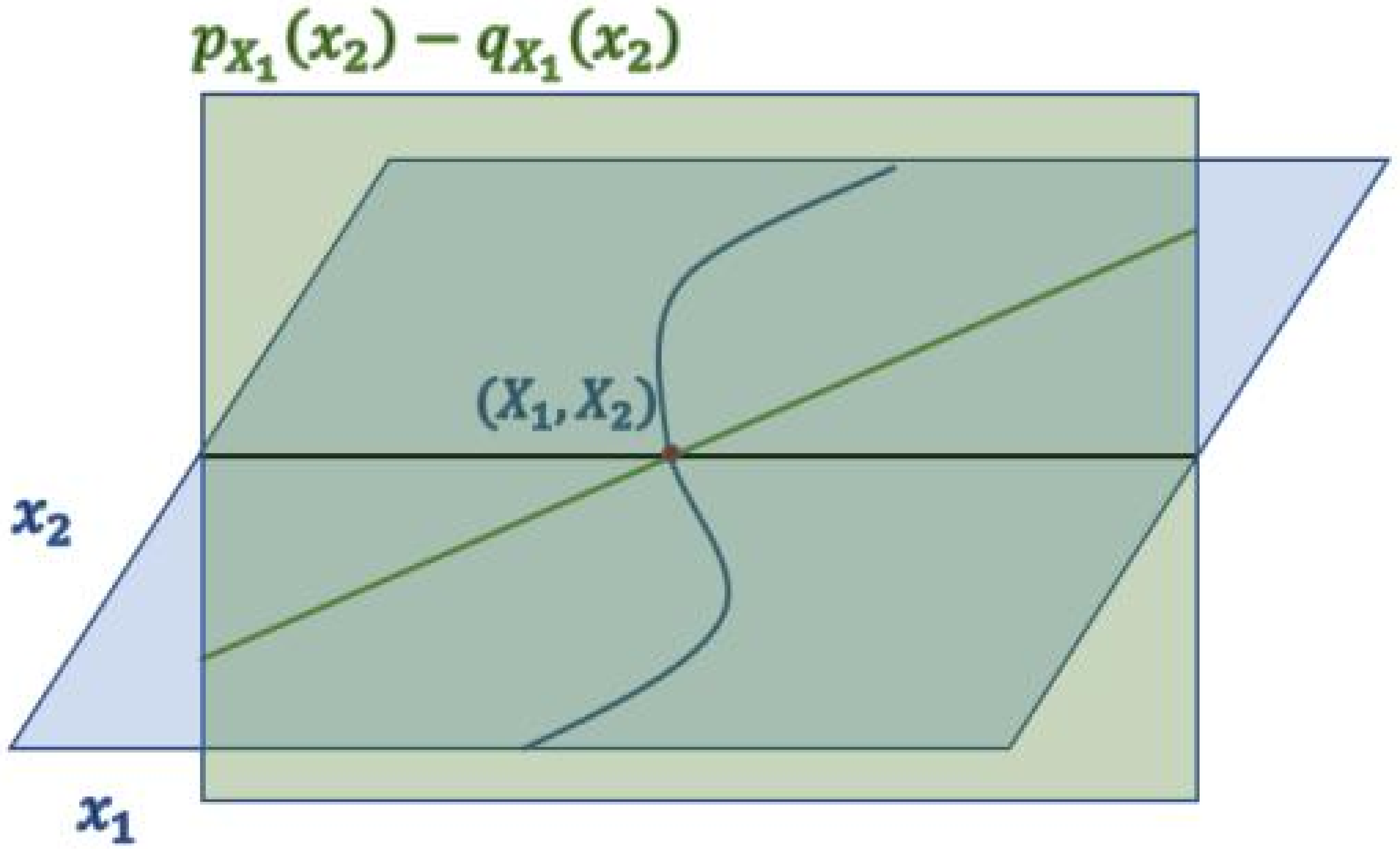
Tsybakov's noise condition with noise exponent κ *or* α:

ASSUMPTION (MA). There exist constants $C_0 > 0$ and $\alpha \geq 0$ such that

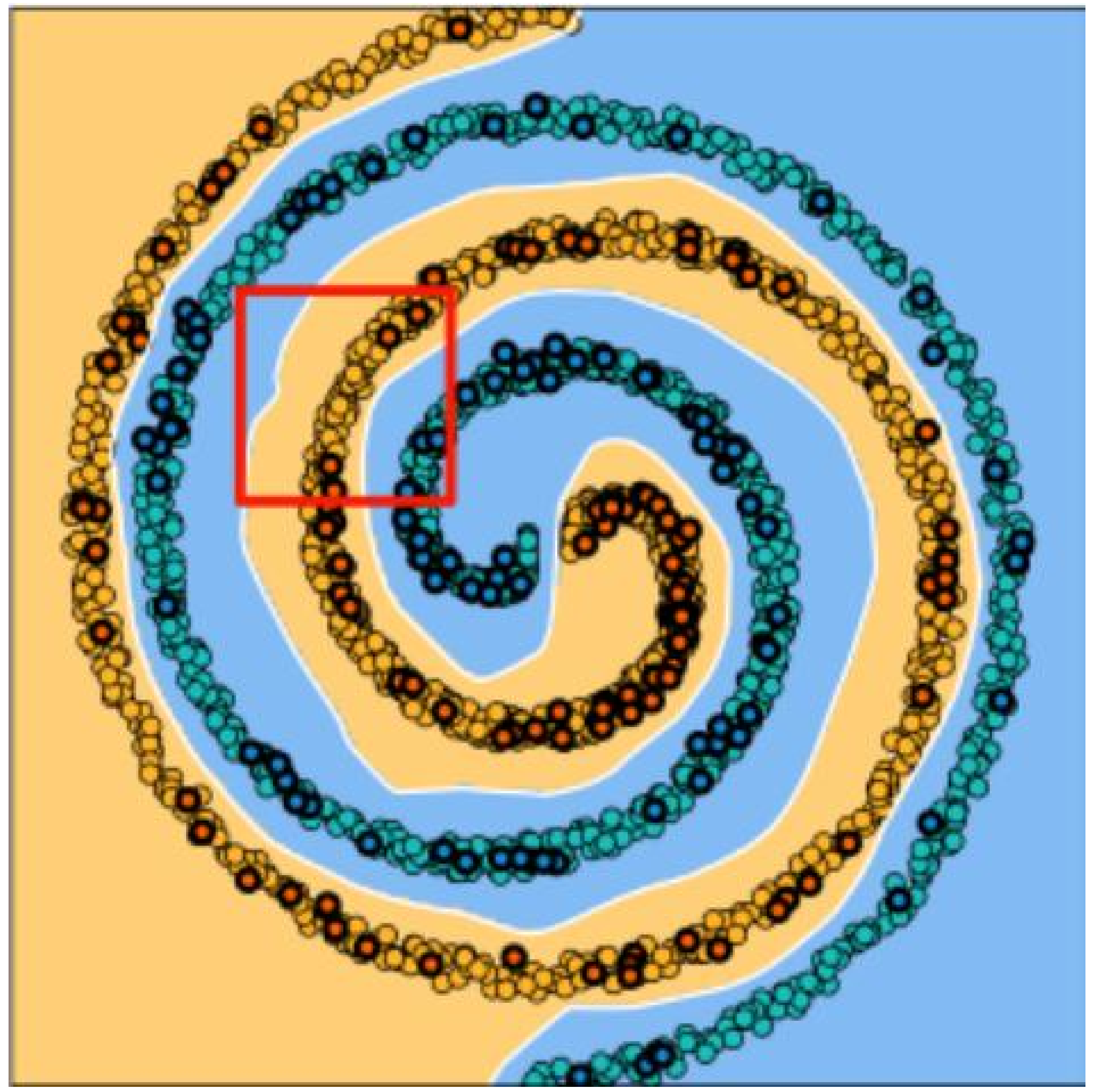
$$P_X(0 < |\eta(X) - 1/2| \leq t) \leq C_0 t^{\alpha} \qquad \forall t > 0.$$

(N) There exists $C, T > 0$ and $\kappa \in [0, \infty]$ such that for any $0 < t < T$

$$\mathbb{Q}(\{x : |p(x) - q(x)| \leq t\}) \leq Ct^{\kappa}.$$



Separated with positive margin γ



Existing Results

With Tsybakov's noise condition:

DNN classifiers

(CAR) On the Conditional Probability:

Assume η to be β-smooth (Hölder)

$$\sup_{P \in \mathcal{P}'_\Sigma} \{\mathbf{E}R(\hat{f}_n) - R(f^*)\} \geq Cn^{-(1+\alpha)\beta/((2+\alpha)\beta+d)}.$$

Optimal up to a log term

(CAD) On the Decision Boundary:

Under the **smooth boundary fragment** assumption with smoothness β

Sub-optimal w.r.t. κ

$$O\left(\frac{1}{n}\right)^{\frac{\beta(\kappa+1)}{\beta(\kappa+2)+(d-1)\kappa}}$$

$$O\left(\frac{\log^3 n}{n}\right)^{\frac{\beta(\kappa+1)}{\beta(\kappa+2)+(d-1)(\kappa+1)}}$$

CAR-NN: Fast Rates with Algorithmic Guarantee

Overview of Results:

Overparametrized ReLU network trained with square loss + gradient descent + weight decay

- **Convergence**: Derived fast convergence rates;
- **Robustness**: When classes are separable, square loss has (adversarial) robustness guarantee;
- **Model Calibration**: Square loss is better-calibrated in theory and in experiments

Modified Square Loss for practical training

- Improved label encoding: one-hot → simplex → better performance

"**Understanding Square Loss in Training Overparametrized Neural Network Classifiers**." *NeurIPS 22 Spotlight*
Joint work with Wenjia Wang (HKUST), Jun Wang (HKUST), Zhenguo Li (Huawei)

CAR-NN: Why Square Loss?

◆ Not bad in practice

Table 7: Vision results, accuracy

Model	Task	train with square loss (%)	train with cross-entropy (%)
TCNN (Bai et al., 2018)	MNIST (acc.)	**97.7**	**97.7**
W-Resnet (Zagoruyko & Komodakis, 2016)	CIFAR-10 (acc.)	95.9	**96.3**
ResNet-50 (He et al., 2016)	ImageNet (acc.)	**76.2**	76.1
	ImageNet (Top-5 acc.)	**93.0**	**93.0**
EfficientNet (Tan & Le, 2019)	ImageNet (acc.)	74.6	**77.0**
	ImageNet (Top-5 acc.)	92.7	**93.3**

Hui, L., & Belkin, M. (2020). Evaluation of neural architectures trained with square loss vs cross-entropy in classification tasks. ICLR 2021.

Table 2: NLP results, accuracy

Model	Task	train with square loss (%)	train with cross-entropy (%)
BERT (Devlin et al., 2018)	MRPC	**83.8**	82.1
	SST-2	**94.0**	93.9
	QNLI	**90.6**	**90.6**
	QQP	**88.9**	**88.9**
LSTM+Attention (Chen et al., 2017)	MRPC	**71.7**	70.9
	QNLI	**79.3**	79.0
	QQP	**83.4**	83.1
LSTM+CNN (He & Lin, 2016)	MRPC	**73.2**	69.4
	QNLI	**76.0**	**76.0**
	QQP	84.3	**84.4**

◆ Explicit Feature modeling

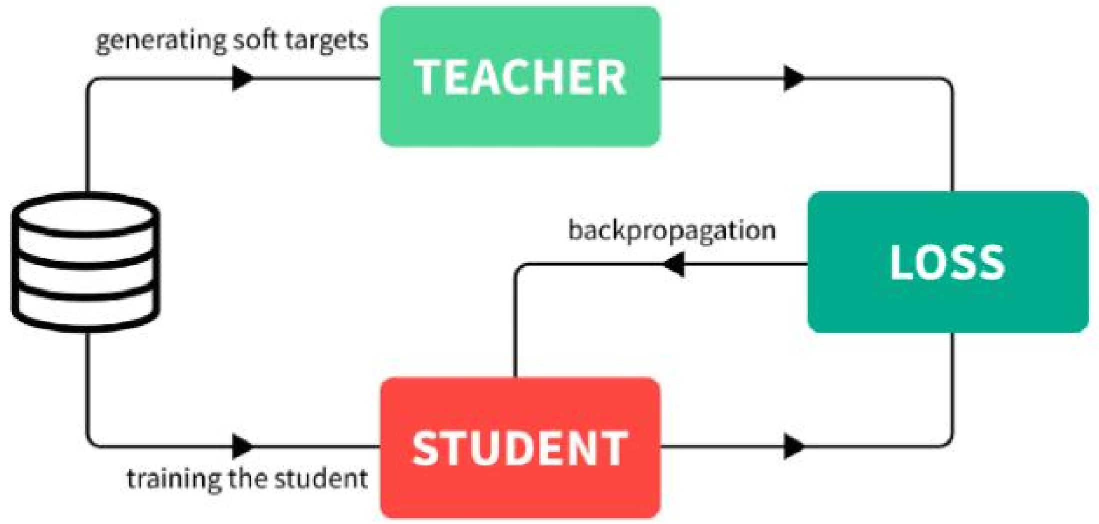
◆ Wide connections

mixup

$$\tilde{x} = \lambda x_i + (1-\lambda)x_j,$$
$$\tilde{y} = \lambda y_i + (1-\lambda)y_j,$$

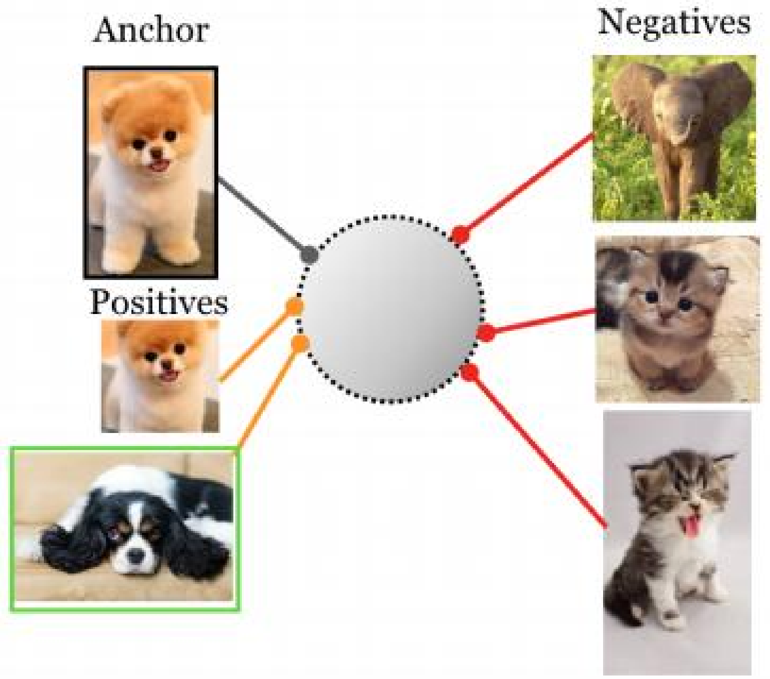
Zhang, Hongyi, et al. "mixup: Beyond empirical risk minimization." *ICLR* 2018.

Knowledge Distillation



Hinton, Vinyals, and Dean. "Distilling the knowledge in a neural network." NIPS 2015

Contrastive Learning



Khosla, Prannay, et al. "Supervised contrastive learning." *NIPS* 2020.

CAR-NN: Model Setup

Model: Overparametrized ReLU Network (in the Neural Tangent Kernel regime)

$$f_{\boldsymbol{W},\boldsymbol{a}}(\boldsymbol{x}) = m^{-1/2} \sum_{r=1}^{m} a_r \sigma(\boldsymbol{W}_r^\top \boldsymbol{x})$$

Training Objective: **Square Loss**

$$l(f_{\boldsymbol{W},\boldsymbol{a}}(\boldsymbol{x}_i), y_i) = (f_{\boldsymbol{W},\boldsymbol{a}}(\boldsymbol{x}_i) - y_i)^2$$

Training Algorithm: Gradient Descent + weight decay (L2 penalty) + early stopping

$$\min_{\boldsymbol{W}} \sum_{i=1}^{n} l(f_{\boldsymbol{W},\boldsymbol{a}}(\boldsymbol{x}_i), y_i) + \mu \mathcal{R}(\boldsymbol{W}, \boldsymbol{a})$$

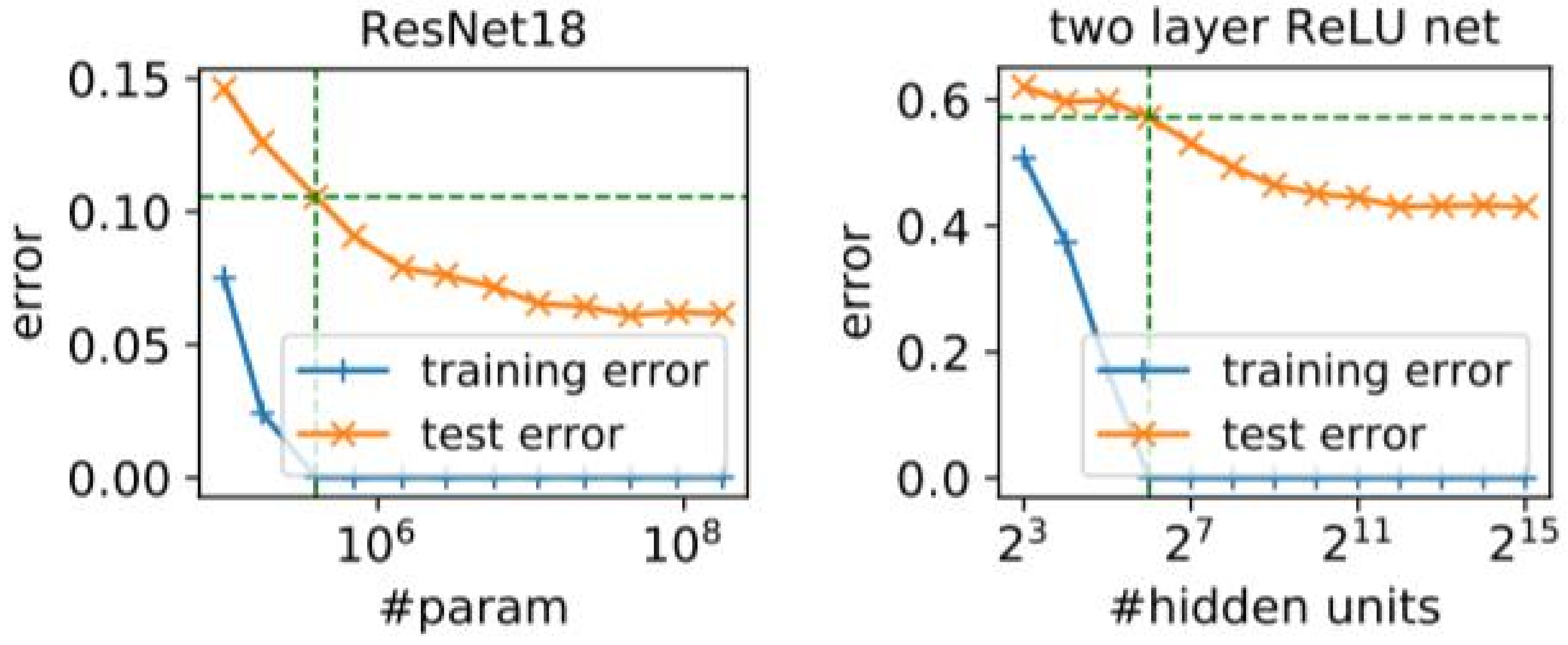
$$\mathcal{R}(\boldsymbol{W}, \boldsymbol{a}) = \|\boldsymbol{W}\|_2^2$$

Interests:

- Accuracy: 0-1 loss excess risk convergence rate
- Robustness: margin when separable
- Model Calibration: estimation of $\eta(x)$

◆ **Why Overparametrization?**

- **Over-parametrization is universal in DL**
- **It helps with optimization and also generalization**



Zhang, Chiyuan, et al. “Understanding deep learning requires rethinking generalization.”
Neyshabur, Behnam, et al. "The role of over-parametrization in generalization of neural networks."

- **Overparametrization networks has a solid theory --- NTK**

Optimization	**Generalization**	**Convergence Rate**
$\Phi(\mathbf{W}(k+1)) \leq \left(1 - \frac{\eta\lambda_0}{2}\right)\Phi(\mathbf{W}(k))$,	$\sqrt{\frac{2\mathbf{y}^\top(\mathbf{H}^\infty)^{-1}\mathbf{y}}{n}}$	$\|f_{\boldsymbol{W}_D(k),\boldsymbol{a}} - f^*\|_2^2 = O_{\mathbb{P}}(n^{-\frac{d}{2d-1}})$.

Arora, et al. "Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks." *ICML* 2019.
Hu, et al. "Regularization Matters: A Nonparametric Perspective on Overparametrized Neural Network." *AISTATS* 2021.

CAR-NN: Assumptions

Data assumption

- The **ground-truth** $\eta(x)$ to be well-behaved (Assumption D.2)
- The **marginal density** of x is assumed to be upper bounded (Assumptions D.4) or both upper and lower bounded (Assumptions D.5)
- Assumption D.2 ensures the problem is not mis-specified.

Model assumption

- The ReLU neural network is to be sufficiently **overparameterized** (with a finite width) plus some regularity conditions on the **GD algorithm** (Assumption D.1)
- The **complexity** of the neural network estimator generated by the GD training is controlled (Assumption D.3)
- Under these assumptions, NN behaves like kernel ridge regression with NTK

Cases Considered:

- General non-separable case
- Separable case with positive margin

T. Hu, W Wang, C Lin, G. Cheng, *Regularization Matters: A Nonparametric Perspective on Overparametrized Neural Network*, AISTATS 2021

CAR-NN: Convergence Rate in the General Case

General non-separable case, fast convergence rate

Theorem 3.1 (Excess risk in the non-separable case)**.** Suppose Assumptions D.1, D.2, and D.4 hold. Assume the conditional probability $\eta(\boldsymbol{x})$ satisfies Tsybakov's noise condition with component κ. Let $\mu \asymp n^{\frac{d-1}{2d-1}}$. Then

$$L(f_{\mathbf{W}(k),\boldsymbol{a}}) = L^* + O_{\mathbb{P}}(n^{-\frac{d(\kappa+1)}{(2d-1)(\kappa+2)}}). \tag{3.1}$$

- The bigger the κ , the faster the rate (can be faster than $1/\sqrt{n}$)
- Under Assumption D.2+D.4, the optimal rate [1] is hypothesized to be $O_{\mathbb{P}}(n^{-\frac{d(\kappa+1)}{d\kappa+4d-2}})$
- Our rate has an extra $(d-1)\kappa$ term in the denominator

$$n^{-\frac{d(\kappa+1)}{(2d-1)(\kappa+2)}} = n^{-\frac{d(\kappa+1)}{(d-1)\kappa+d\kappa+4d-2}}$$

- In another work [2], the rate from plug-in kernel estimate is $O_{\mathbb{P}}(n^{-\frac{\kappa+1}{\kappa+3+d}})$, which is slower than ours
- No similar results for NN classifiers under cross-entropy

[1] Audibert and Tsybakov. Fast learning rates for plug-in classifiers. The Annals of Statistics, 2007.
[2] Kohler and Krzyzak. On the rate of convergence of local averaging plug-in classification rules under a margin condition. IEEE Transactions on Information Theory, 2007.

CAR-NN: Convergence Rate in the Separable Case

Separable case with positive margin, super fast convergence rate

Theorem 3.2 (Generalization error in the separable case). Suppose Assumptions D.1, D.3, and D.5 hold. Let $\mu = o(1)$. There exist positive constants C_1, C_2 such that the misclassification rate is 0% with probability at least $1 - \delta - C_1 \exp(-C_2 n)$, and δ can be arbitrarily small[2] by enlarging the neural network's width.

Lemma 3.5 (Tsybakov's noise condition under Gaussian noises). Let the margin be $2\gamma > 0$, the noise be $N(0, v^2 \boldsymbol{I}_d)$. Then there exist some constants $T, C > 0$ such that

$$P_X(|2\widetilde{\eta}_v(X) - 1| < t) \leq (Cv^2/\gamma) \exp(-\gamma^2/(2v^2))t, \forall t \in (0, T).$$

Theorem 3.6 (Exponential convergence rate). Suppose the classes are separable with margin $2\gamma > 0$. No matter how complicated $\Omega_1 \cup \Omega_2$ are, the excess risk of the overparameterized neural network classifier satisfying Assumptions D.1 and D.4 has the rate $O_{\mathbb{P}}(e^{-n\gamma/7})$.

CAR-NN: Label Coding for Multi-Class

Square loss is not inferior to cross entropy，could be even better!

Table 7: Vision results, accuracy

Model	Task	train with square loss (%)	train with cross-entropy (%)
TCNN (Bai et al., 2018)	MNIST (acc.)	**97.7**	**97.7**
W-Resnet (Zagoruyko & Komodakis, 2016)	CIFAR-10 (acc.)	95.9	**96.3**
ResNet-50 (He et al., 2016)	ImageNet (acc.)	**76.2**	76.1
	ImageNet (Top-5 acc.)	**93.0**	**93.0**
EfficientNet (Tan & Le, 2019)	ImageNet (acc.)	74.6	**77.0**
	ImageNet (Top-5 acc.)	92.7	**93.3**

Dataset	#classes	k	M
MRPC	2	1	1
SST-2	2	1	1
QNLI	2	1	1
QQP	2	1	1
TIMIT (CER)	27	1	1
TIMIT (WER)	42	1	15
WSJ	52	1	15
Librispeech	1000	15	30
MNIST	10	1	1
CIFAR-10	10	1	1
ImageNet	1000	15	30

A trick for multi-class classification

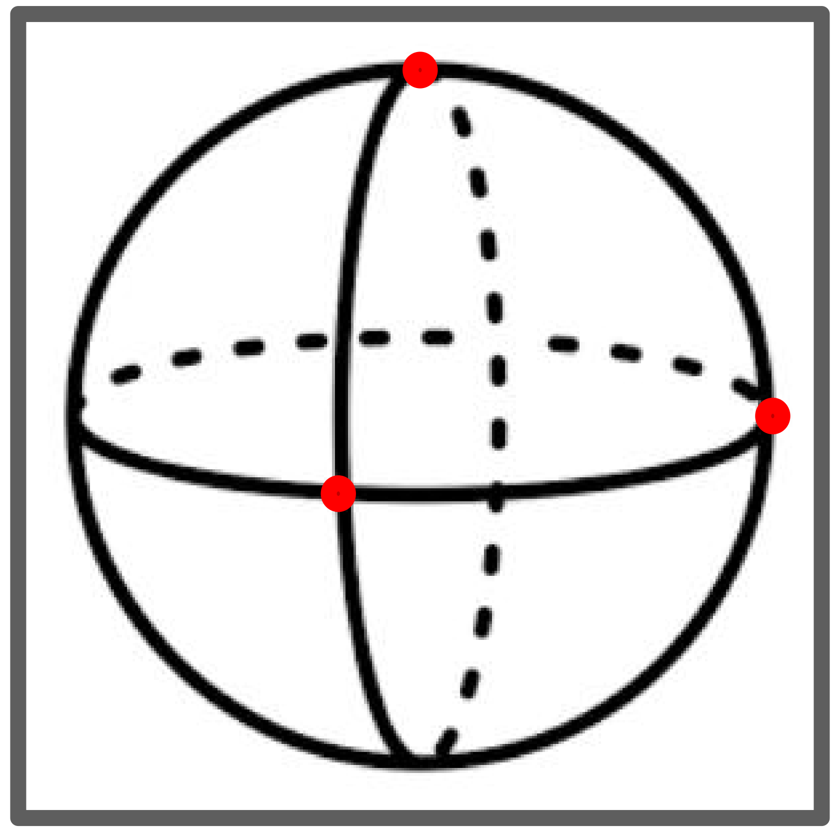
$$l = \frac{1}{C}\left((f_c(x) - 1)^2 + \sum_{i=1, i\neq c}^{C} f_i(x)^2\right) \quad \Rightarrow \quad l_s = \frac{1}{C}\left(k * (f_c(x) - M)^2 + \sum_{i=1, i\neq c}^{C} f_i(x)^2\right).$$

Why square loss struggles when the number of classes is large?

Hui, L., & Belkin, M. Evaluation of neural architectures trained with square loss vs cross-entropy in classification tasks. ICLR 2021.

CAR-NN: Modified Label Coding for Multi-Class

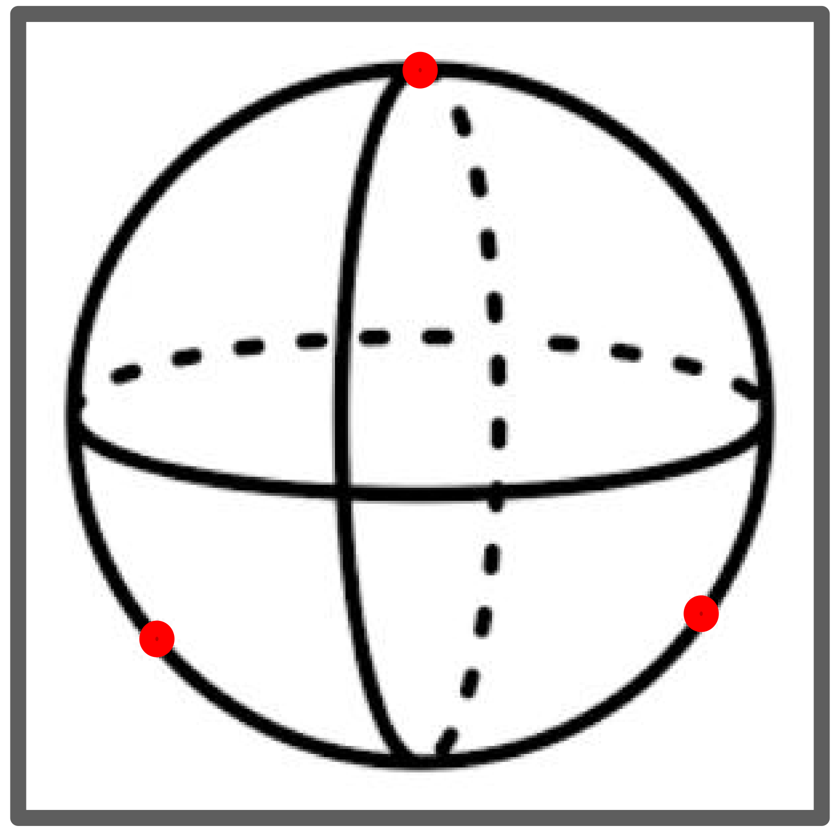
Modify label encoding from one-hot → simplex



Proposition 3.7 (Conditional probability). Let $f^* : \Omega \to \mathbb{R}^K$ minimize the mean square error $\mathbb{E}_X(f^*(X) - \boldsymbol{v}_y)^2$, where $\boldsymbol{v}_y$ is the simplex coding vector of label y. Then

$$\eta_k(\boldsymbol{x}) := \mathbb{P}\left(y = k|\boldsymbol{x}\right) = \left((K-1)f^*(\boldsymbol{x})^\top \boldsymbol{v}_k + 1\right)/K. \tag{3.2}$$

Experiments: ResNet-18 on CIFAR-10 **+0.5%**, ResNet-50 onCIFAR-100 **+6%**



Coincides with Supervised Contrastive Learning (Khosla, Prannay, et al.):

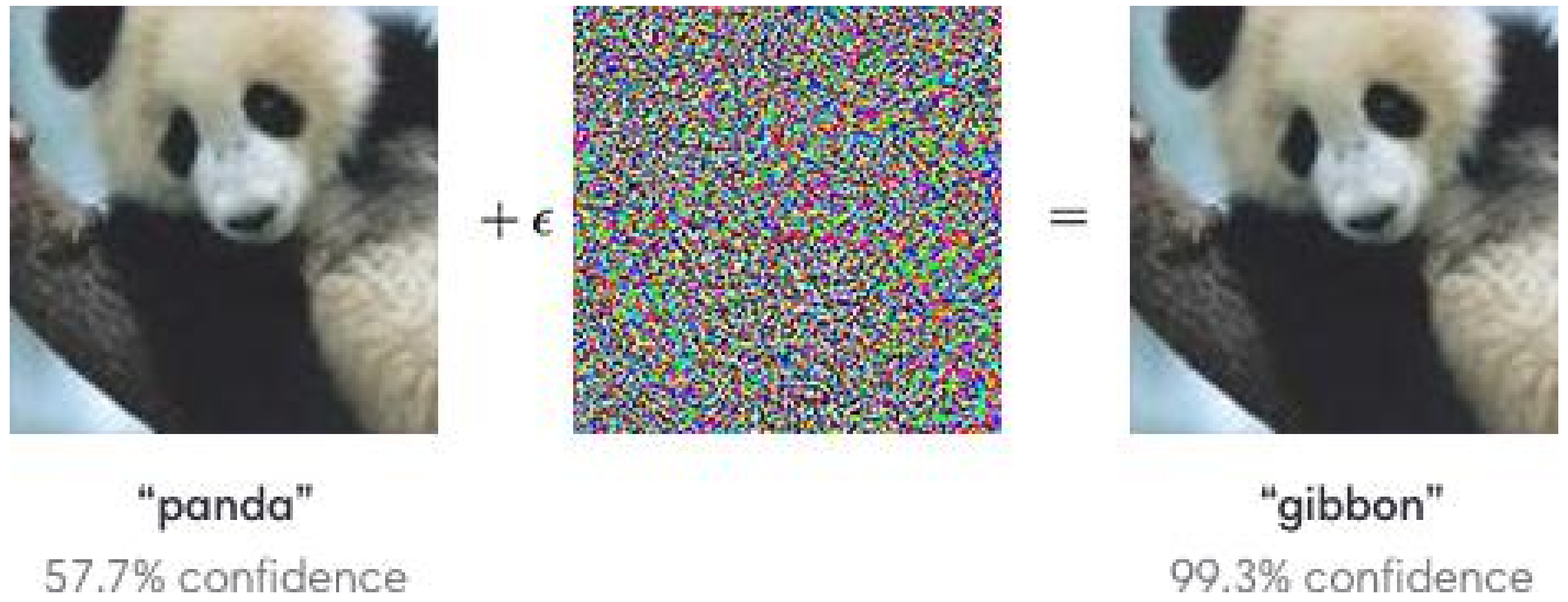
Dataset	SimCLR[3]	Cross-Entropy	Max-Margin [32]	SupCon
CIFAR10	93.6	95.0	92.4	**96.0**
CIFAR100	70.7	75.3	70.5	**76.5**
ImageNet	70.2	78.2	78.0	**78.7**

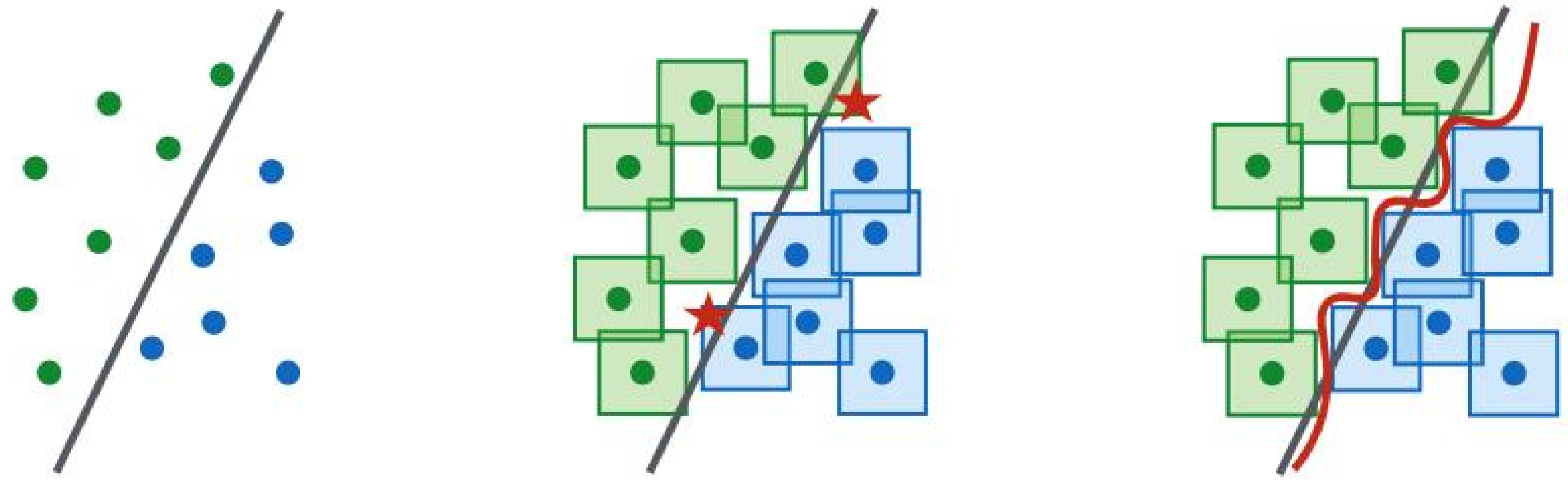
Table 2: Top-1 classification accuracy on ResNet-50 [17] for various datasets. We compare cross-entropy training, unsupervised representation learning (SimCLR [3]), max-margin classifiers [32] and SupCon (ours). We re-implemented and tuned hyperparameters for all baseline numbers except margin classifiers where we report published results. Note that the CIFAR-10 and CIFAR-100 results are from our PyTorch implementation and ImageNet from our TensorFlow implementation.

CAR-NN: Robustness in the Separable Case

When separable with positive margin, square loss has (adversarial) robustness guarantee

Measurement: size of the empirical margin





Theorem 3.3 (Robustness in the separable case)**.** Suppose the assumptions of Theorem 3.2 are satisfied. Let $\mu = o(1)$. Then there exist positive constants C, C_1, C_2 such that $\min_{\boldsymbol{x}\in\mathcal{D}_T, \boldsymbol{x}'\in\Omega_1\cup\Omega_2} \|\boldsymbol{x}-\boldsymbol{x}'\|_2 \geq C$, and the misclassification rate is 0% with probability at least $1-\delta-C_1\exp(-C_2 n)$ for all n, where $\mathcal{D}_T$ is the decision boundary, and δ is as in Theorem 3.2.

- The GD implicit bias under CE is maximize training margin
- The margin in our theorem is on the **population level**

Lyu, Kaifeng, and Jian Li. "Gradient Descent Maximizes the Margin of Homogeneous Neural Networks." *ICLR* 2019.

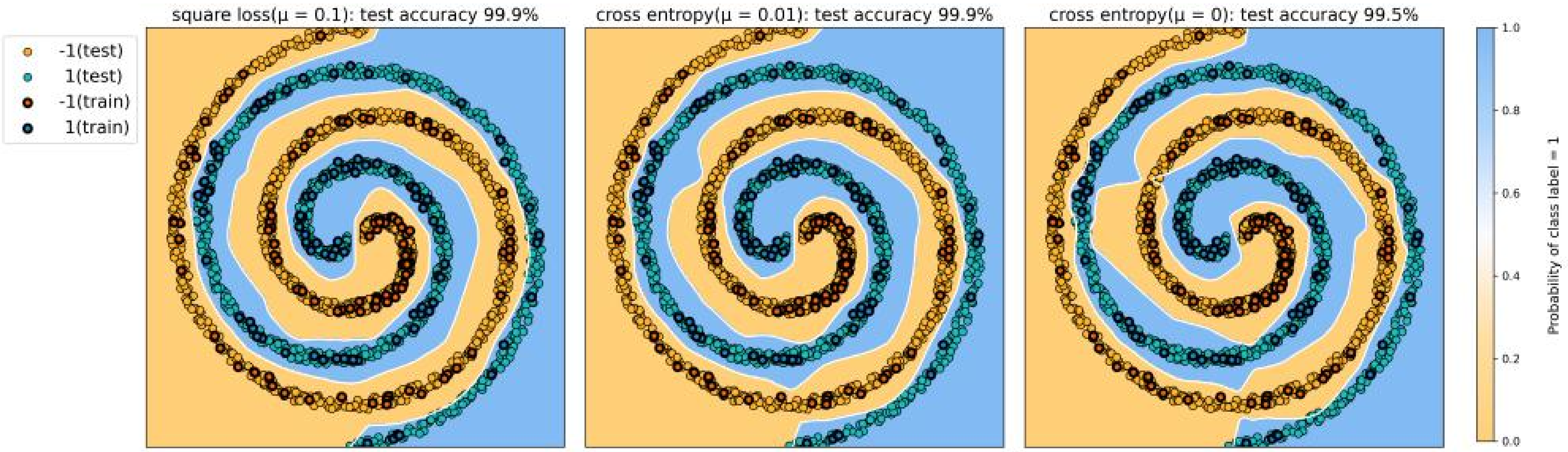
CAR-NN: Numerical Experiments - Accuracy and Robustness

Table 1: Test accuracy on CIFAR datasets. Average accuracy larger than 0 but less than 0.1 is denoted as 0^* without standard deviation.

Dataset	Network	Loss	Clean acc %	*PGD-100 (l_∞-strength)*			*AutoAttack (l_∞-strength)*		
				2/255	4/255	8/255	2/255	4/255	8/255
CIFAR-10	ResNet-18	CE	**95.15 (0.11)**	8.81 (1.61)	0.65 (0.24)	0	2.74 (0.09)	0	0
		SL	95.04 (0.07)	**30.53 (0.92)**	**6.64 (0.67)**	**0.86 (0.24)**	**4.10 (0.50)**	**0^***	0
	WRN-16-10	CE	93.94 (0.16)	1.04 (0.10)	0	0	0.33 (0.06)	0	0
		SL	**95.02 (0.11)**	**37.47 (0.61)**	**23.16 (1.28)**	**7.88 (0.72)**	**5.37 (0.50)**	**0^***	0
CIFAR-100	ResNet-50	CE	**79.82 (0.14)**	2.31 (0.07)	0^*	0	0.99 (0.10)	0^*	0
		SL	78.91 (0.14)	**13.76 (1.30)**	**4.63 (1.20)**	**1.21 (0.80)**	**3.67 (0.60)**	**0.16 (0.05)**	0
	WRN-16-10	CE	77.89 (0.21)	0.83 (0.07)	0^*	0	0.42 (0.07)	0	0
		SL	**79.65 (0.15)**	**6.48 (0.40)**	**0.42 (0.04)**	**0^***	**2.73 (0.20)**	**0^***	0

Performance on CIFAR-10 dataset for ResNet-18 under standard PGD adversarial training.

Loss	Acc (%)	PGD steps	Strength(l_∞)	AutoAttack
CE	86.87	3	8/255	37.08
	84.50	7	8/255	41.88
SL	**87.31**	3	8/255	**40.46**
	84.52	7	8/255	**44.76**



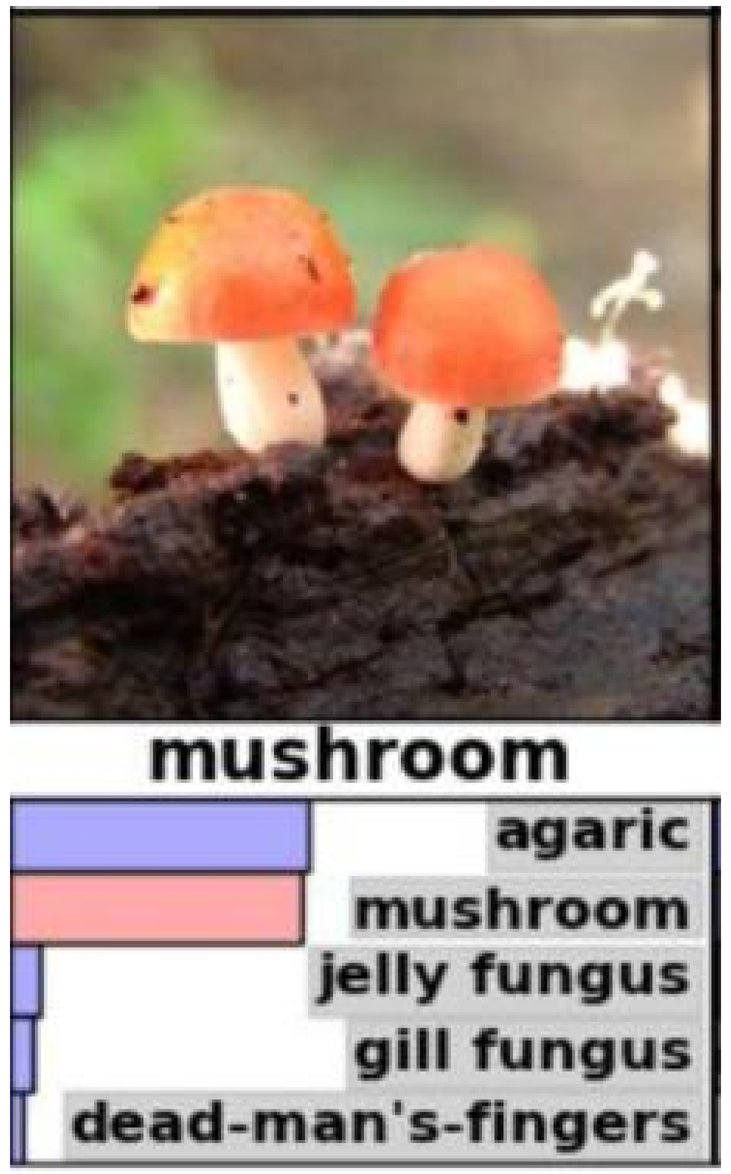
CAR-NN: Model Calibration

Square loss is better-calibrated in theory and in experiments

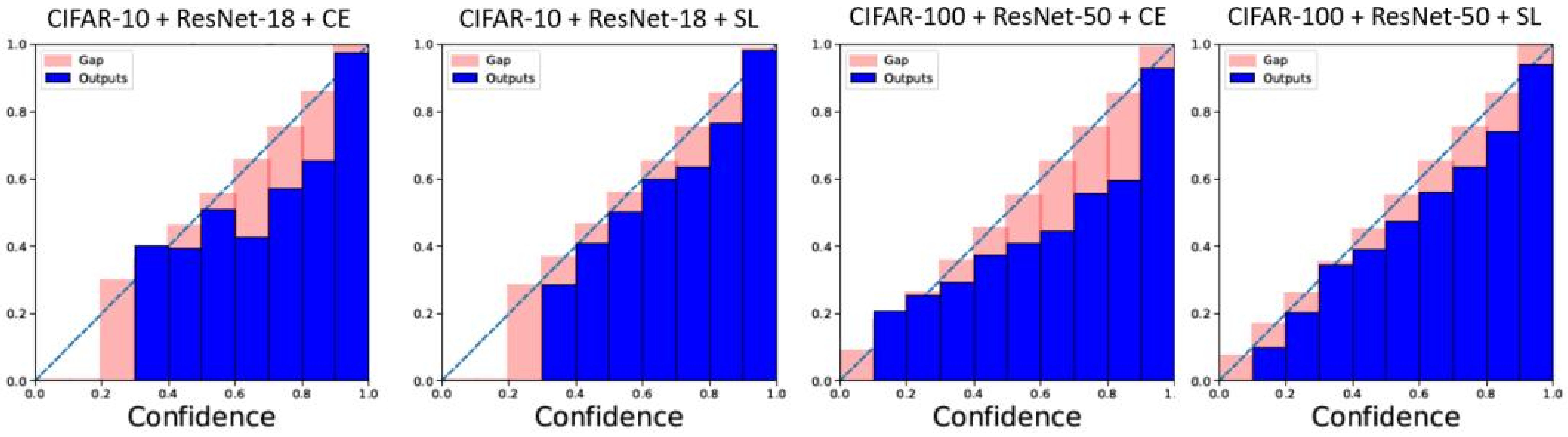
Measurement: expected calibration error, or $||\eta - \hat{\eta}||_\infty$

Theorem 3.4 (Calibration error). Suppose Assumptions D.1-D.4 are fulfilled. Let $\mu \asymp n^{\frac{d-1}{2d-1}}$. Then

$$\left\|(f_{\boldsymbol{W}(k),\boldsymbol{a}} + 1)/2 - \eta\right\|_{L_\infty} = O_{\mathbb{P}}(n^{-1/(4d-2)}).$$



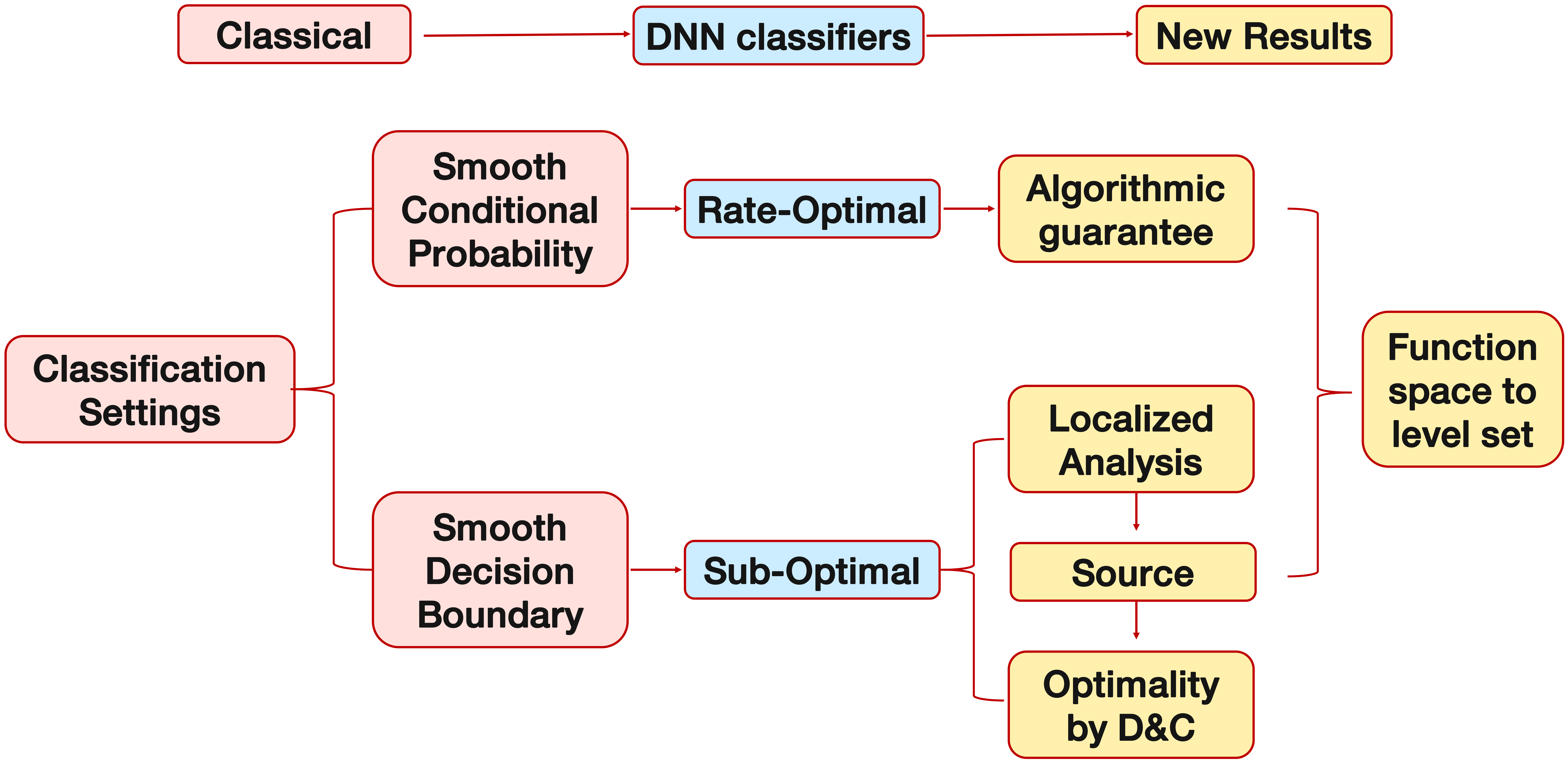
Experiments:



Square Loss:
$\hat{\eta} = (f_w + 1)/2$

Cross Entropy:
$\hat{\eta} = e^{f_w}/(1 + e^{f_w})$

Outline



- *Understanding Square Loss in Training Overparameterized Neural Network Classifiers*, **NeurIPS 2022 Spotlight**, with Wenjia Wang, Jun Wang, Zhenguo Li
- *Minimax Optimal Deep Neural Network Classifiers Under Smooth Decision Boundary*, arXiv, with Ruiqi Liu, Zuofeng Shang, Guang Cheng
- *Exact Count of Boundary Pieces of ReLU Classifiers:Towards the Proper Complexity Measure for Classification*, **UAI 2023**, with Pawel Piwek, Adam Klukowski

CAD-NN: Recap

(CAD) On the Decision Boundary:

$$\mathcal{H}(\varepsilon, \mathcal{G}, d_\Delta) \le A_* \varepsilon^{-\rho} \qquad \forall \varepsilon > 0,$$

DNN classifiers

Under the **smooth boundary fragment** assumption

with smoothness β

Sub-optimal w.r.t. κ

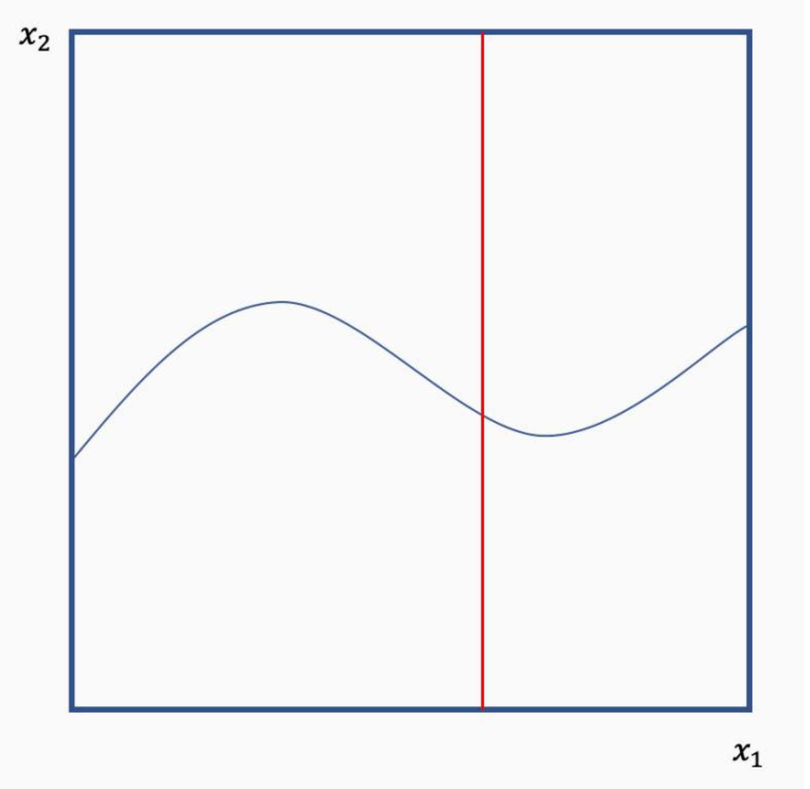
$$O\left(\frac{1}{n}\right)^{\frac{\beta(\kappa+1)}{\beta(\kappa+2)+(d-1)\kappa}}$$

$$O\left(\frac{\log^3 n}{n}\right)^{\frac{\beta(\kappa+1)}{\beta(\kappa+2)+(d-1)(\kappa+1)}}$$

Smooth Boundary Fragment: Function as boundary

For $d \ge 2$, let $\boldsymbol{x}_{-d} = (x_1, \cdots, x_{d-1})$. The smooth boundary fragment setting assumes the optimal set G^* to have the form

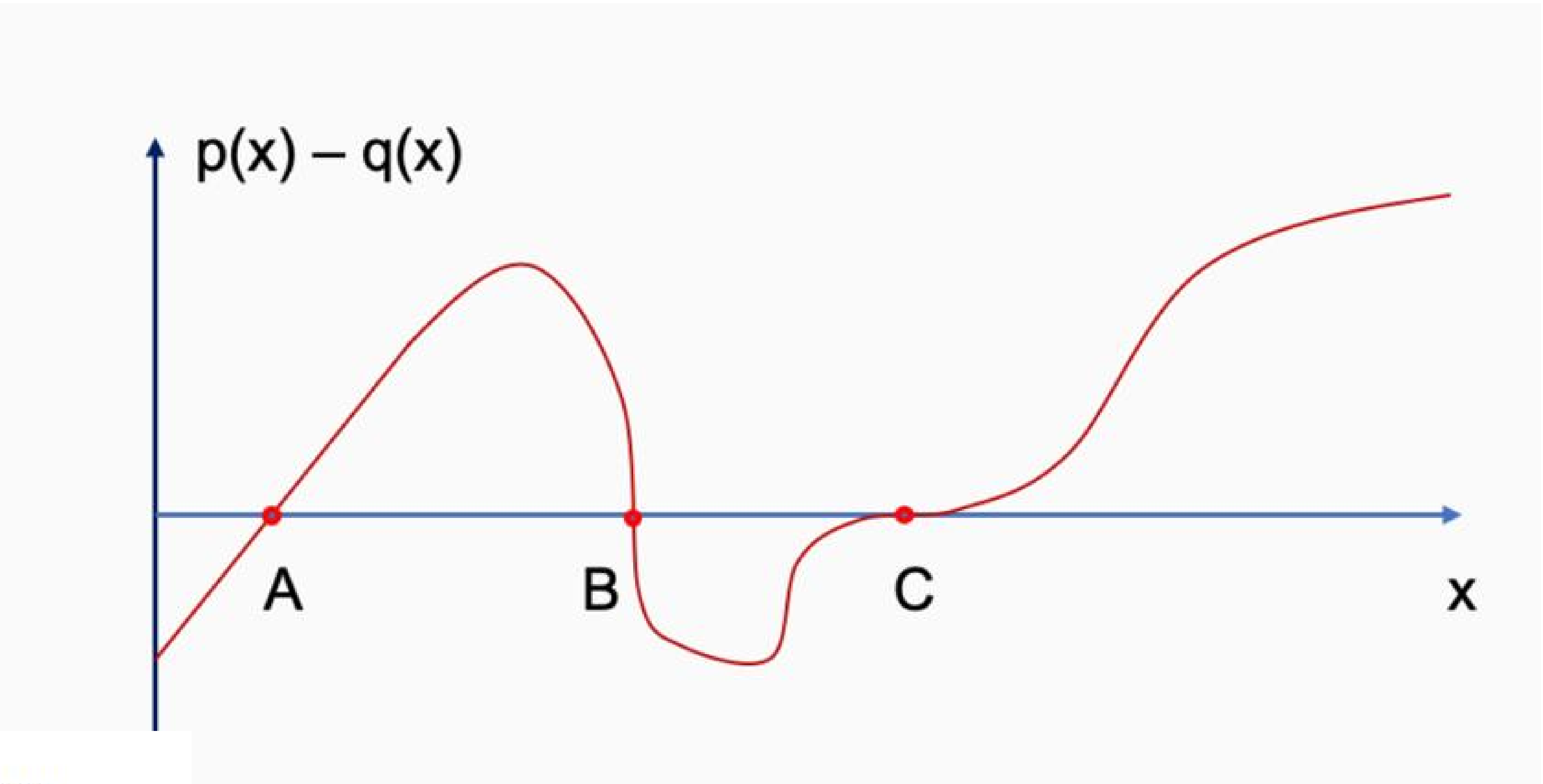
$$G_{f^*} := \{\boldsymbol{x} \in \mathbb{R}^d : f^*(\boldsymbol{x}_{-d}) - x_d \ge 0, f^* \in \mathcal{H}(d, \beta)\}. \tag{2.2}$$



Other ways: By ReLU network

T. Hu, Z. Shang, G. Cheng, *Sharp Rate of Convergence for Deep Neural Network Classifiers Under the Teacher-Student Setting*

CAD-NN: Source of Sub-optimality



Inconsistency of κ along the decision boundary

Decomposition of the excess risk:

$$\textit{excess risk} \approx (\textit{approx error})^{1+1/\kappa} + (\textit{stochastic error})^{1-\frac{1}{\kappa+2}}$$

(N) There exists constants $c, T > 0$ and $\kappa \in [0, \infty]$ such that for any $0 \le t \le T$,

$$\mathbb{Q}\left(\{\boldsymbol{x} : |p(\boldsymbol{x}) - q(\boldsymbol{x})| \le t\}\right) \le ct^{\kappa}.$$

(N^+) There exist constants $c_1, T > 0$ and $\kappa \in [0, \infty]$ such that for any $0 \le t \le T$,

$$\mathbb{Q}\left(\{\boldsymbol{x} \in G : |p(\boldsymbol{x}) - q(\boldsymbol{x})| \le t\}\right) \ge c_1 t^{\kappa}$$

holds for any positive-measure set $G \subset \mathcal{X}$ containing the decision boundary, i.e., $\partial G^* \cap G^\circ$ is not empty.

Lemma 3.1. (Informal) Under assumptions (N) and the smooth boundary fragment assumption (2.2), if we further assume (N^+), then the empirical 0-1 loss minimizer within a ReLU DNN family with proper size achieves the optimal 0-1 loss excess risk convergence rate of $n^{-\frac{\beta(\kappa+1)}{\beta(\kappa+2)+(d-1)\kappa}}$.

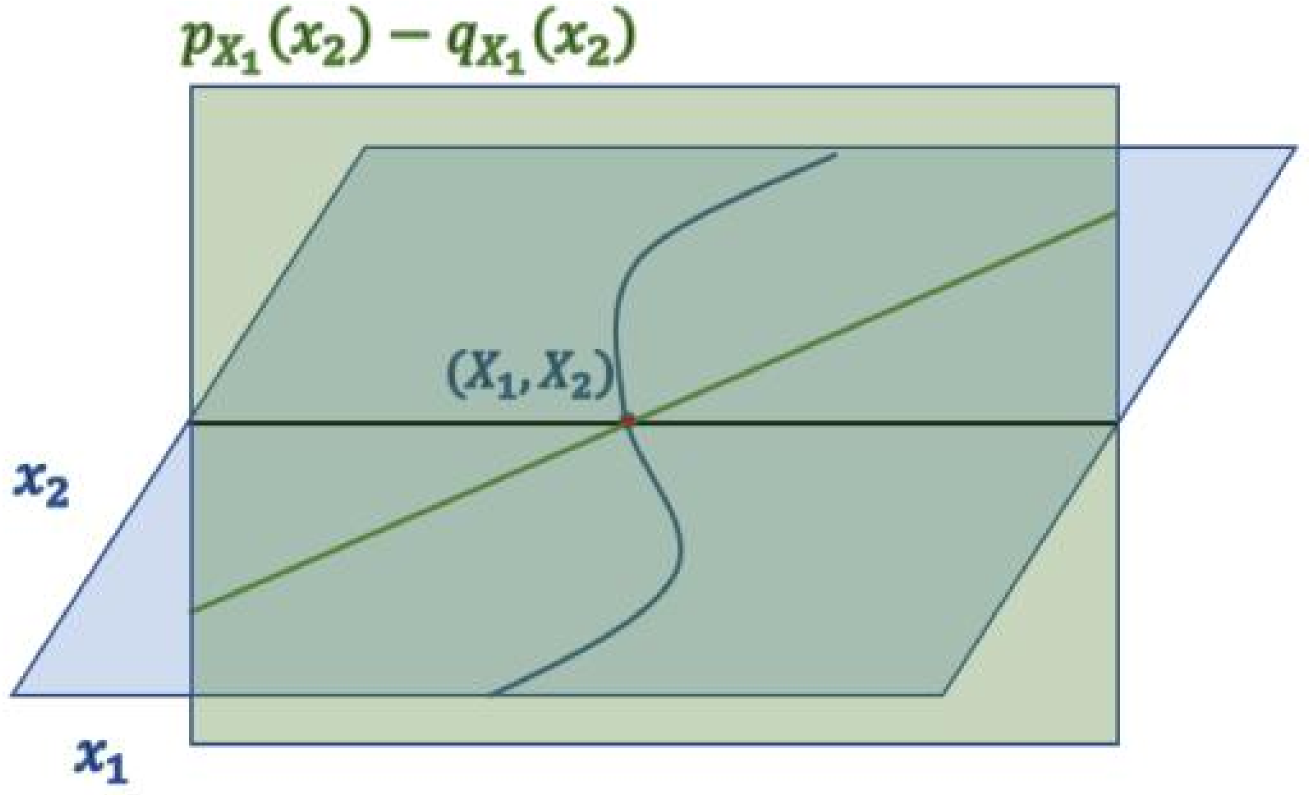
Matches the lower bound!

CAD-NN: Localized Analysis

Localized Tsybakov's noise condition

$$m_{\boldsymbol{x}_{-d}}(t) := |p((\boldsymbol{x}_{-d}, f^*(\boldsymbol{x}_{-d}) + t)) - q((\boldsymbol{x}_{-d}, f^*(\boldsymbol{x}_{-d}) + t))|,$$

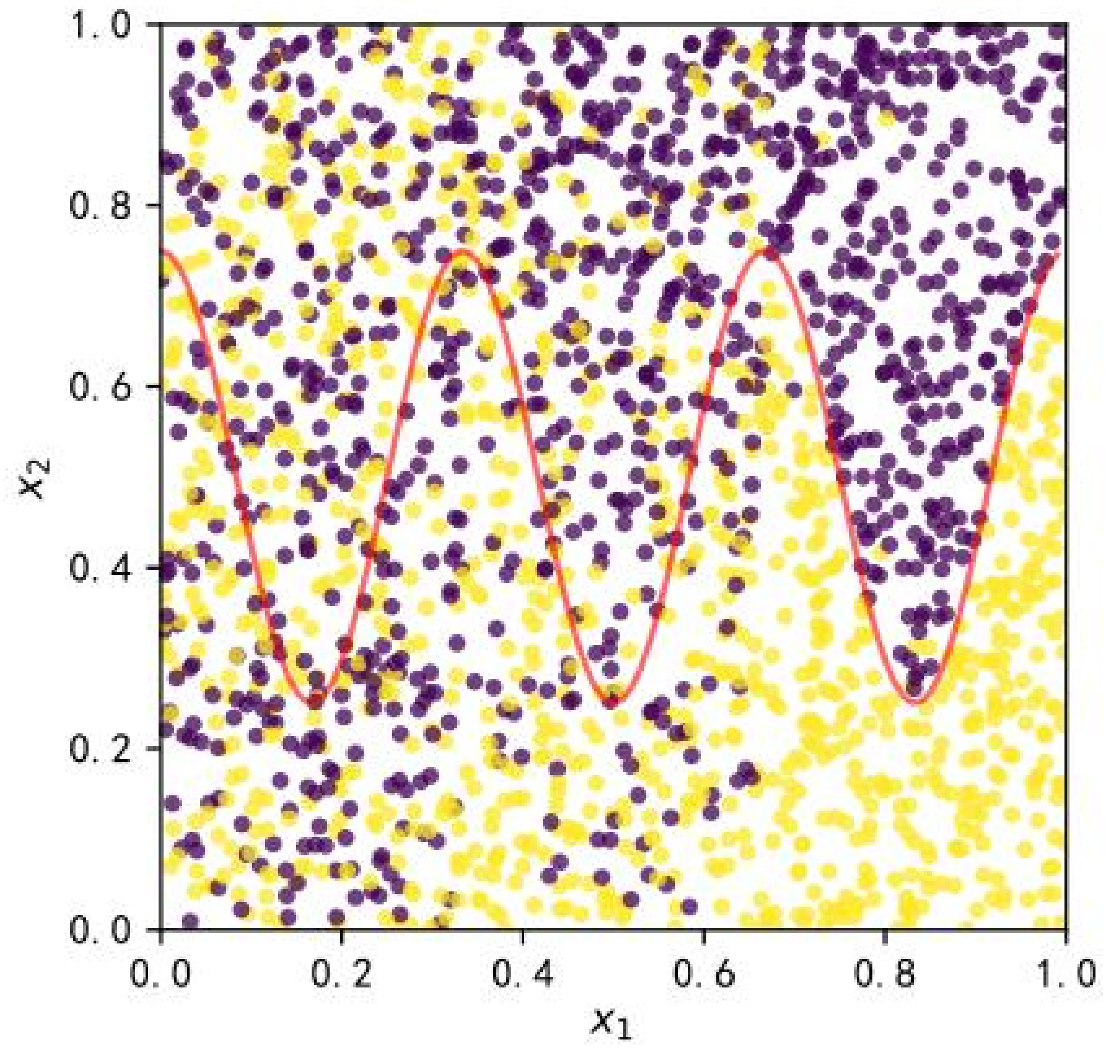
$$K(\boldsymbol{x}) = \sup\{k \geq 0 : \lim_{t \to 0} \frac{m_{\boldsymbol{x}}(t)}{|t|^{1/k}} > 0\}.$$

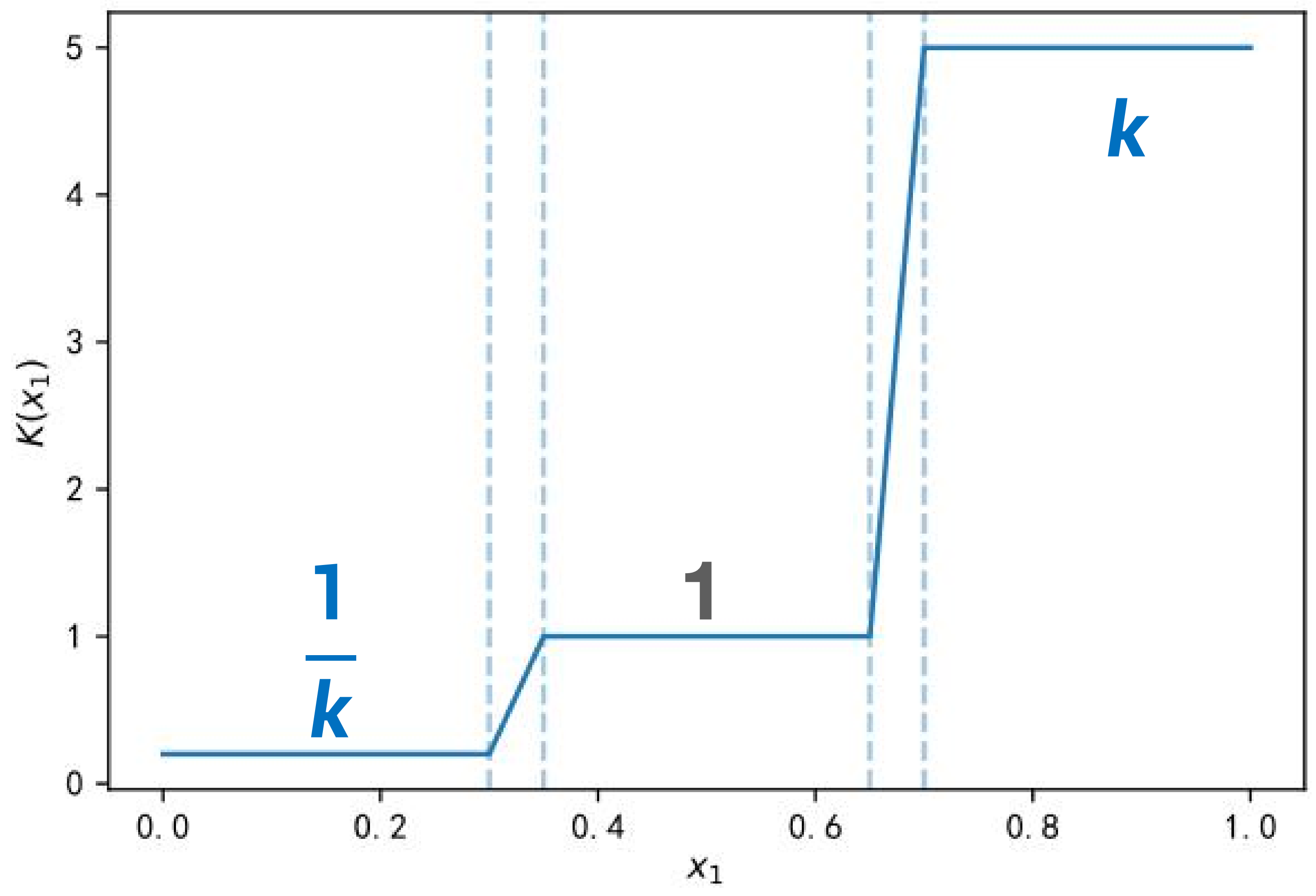


2D Example: Let $x \in [0,1]^2$ be uniformly distributed, i.e., $p(x) + q(x) \equiv 2$.

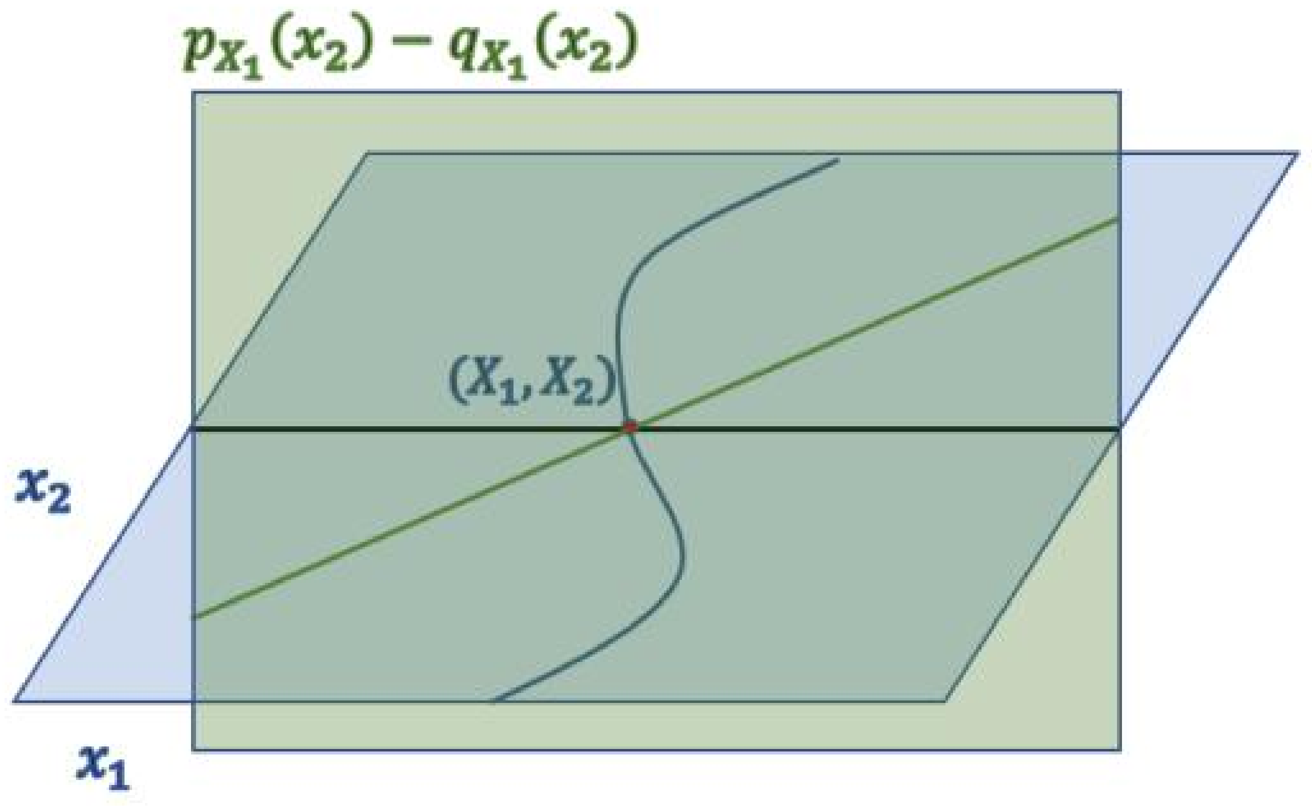
Decision boundary: $x_2 = f^*(x_1) = \frac{1}{4}\cos(6\pi x_1) + \frac{1}{2}$. Let $\delta(x) = \frac{4}{3}(x_2 - f^*(x_1))$, which ranges from -1 to 1.

By setting $2\eta(\boldsymbol{x}) - 1 = \mathrm{sign}(\delta(\boldsymbol{x})) \cdot \delta(\boldsymbol{x})^{K(\boldsymbol{x})}$ will allow us to specify K(x) freely.





CAD-NN: Localized Analysis



Localized Tsybakov's noise condition

$$m_{\boldsymbol{x}_{-d}}(t) := |p((\boldsymbol{x}_{-d}, f^*(\boldsymbol{x}_{-d}) + t)) - q((\boldsymbol{x}_{-d}, f^*(\boldsymbol{x}_{-d}) + t))|,$$

$$K(\boldsymbol{x}) = \sup\{k \geq 0 : \lim_{t \to 0} \frac{m_{\boldsymbol{x}}(t)}{|t|^{1/k}} > 0\}.$$

(M1) There exists $\epsilon_0 > 0$ small enough and a constant $0 < C_{\epsilon_0} < \infty$ such that for all $\boldsymbol{x} \in \partial G^*$ and any $0 < t < \epsilon_0$,

$$\frac{1}{C_{\epsilon_0}} \leq \frac{m_{\boldsymbol{x}}(t)}{|t|^{1/K(\boldsymbol{x})}} \leq C_{\epsilon_0}.$$

Lemma 3.3. Denote $\kappa^- = \inf_{\boldsymbol{x} \in \partial G^*} K(\boldsymbol{x})$ and $\kappa^+ = \sup_{\boldsymbol{x} \in \partial G^*} K(\boldsymbol{x})$. Then condition (M1) implies that (N) holds with $\kappa = \kappa^-$ and (N$^+$) holds with $\kappa = \kappa^+$.

Theorem 3.4. Under the smooth boundary fragments setting (2.2) with smoothness β. Assume condition (M1) and let $\kappa^- = \inf_{\boldsymbol{x} \in \partial G^*} K(\boldsymbol{x})$. For any function space $\mathcal{F}$, the 0-1 loss excess risk has the following lower bound,

$$\inf_{\widehat{f} \in \mathcal{F}} \sup_{G^* \in \mathcal{G}_\beta^*} \mathbb{E}[\mathcal{E}(\widehat{f}, G^*)] \gtrsim \left(\frac{1}{n}\right)^{\frac{\beta(\kappa^- + 1)}{\beta(\kappa^- + 2) + (d-1)\kappa^-}}.$$

CAD-NN: Localized Analysis

Localized Convergence Analysis

Theorem 3.5. Under assumption (M1), further assume that for some $\boldsymbol{j}_{-d} \in J_M$, $\kappa^- \leq K(\boldsymbol{x}) \leq \kappa^+$ for all $\boldsymbol{x} \in D_{\boldsymbol{j}_{-d}}$. Let $\widetilde{\mathcal{F}}_n$ be a ReLU DNN family[1] with size in the order of

$$\widetilde{N}_n \widetilde{L}_n \asymp n^{\frac{\kappa^+(\kappa^-+1)(d-1)/2}{(\kappa^-+2)(\kappa^++1)\beta+(d-1)\kappa^+(\kappa^-+1)}} \cdot \log^2(n).$$

Let the empirical 0-1 loss minimizer be

$$\widehat{f}_{n,\boldsymbol{j}_{-d}} := \underset{f \in \widetilde{\mathcal{F}}_n}{\operatorname{argmin}} R_{n,\boldsymbol{j}_{-d}}(f). \tag{3.1}$$

Then the 0-1 loss excess risk satisfies

$$\sup_{G^* \in \mathcal{G}^*_\beta} \mathbb{E}(R_{\boldsymbol{j}_{-d}}(\widehat{f}_{n,\boldsymbol{j}_{-d}}) - R_{\boldsymbol{j}_{-d}}(G^*)) = \widetilde{O}\left(n^{-\frac{(\kappa^-+1)\beta}{(\kappa^-+2)\beta+\left(\frac{\kappa^-+1}{\kappa^++1}\right)(d-1)\kappa^+}}\right),$$

where $\widetilde{O}(\cdot)$ hides the $\log(n)$ terms.

If $\kappa^+ = \kappa^-$, optimal rate!

If $\kappa^+ = \infty$, recovers existing fast rate

CAD-NN: Divide-and-Conquer

Global Convergence Analysis

(M2) $K(\boldsymbol{x})$ is α-Hölder continuous for some $0 < \alpha \leq 1$, i.e. there exists constant C_K such that for any $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \partial G^*$,

$$|K(\boldsymbol{x}_1) - K(\boldsymbol{x}_2)| \leq C_K \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2^{\alpha}.$$

Theorem 3.9. Under the smooth boundary fragments setting (2.2), assume conditions (M1,M2). Denote $\kappa^- = \inf_{\boldsymbol{x}\in[0,1]^d} K(\boldsymbol{x})$, $\kappa^+ = \sup_{\boldsymbol{x}\in[0,1]^d} K(\boldsymbol{x})$. Let $\mathcal{F}_n$ be a ReLU DNN family with proper architectures specified in Section 3.3 and size constraint

$$N_n L_n = O\left(n^{\frac{\kappa^+(d-1)/2}{(\kappa^+ + 2)\beta + (d-1)\kappa^+}} \cdot \log^{d+1}(n)\right).$$

Then, with probability tending to one, the empirical 0-1 loss minimizer within $\mathcal{F}_n$ satisfies

$$\inf_{\widehat{f}_n \in \mathcal{F}_n} \sup_{G^* \in \mathcal{G}^*_\beta} \mathbb{E}(R(\widehat{f}_n) - R(G^*)) = \widetilde{O}\left(n^{-\frac{(\kappa^- + 1)\beta}{(\kappa^- + 2)\beta + (d-1)\kappa^-}}\right).$$

CAD-NN: Curse-of-Dimensionality

Compositional Smoothness Structure: Effective smoothness β^* and effective dimension d^*

$$f^* = h_l \circ h_{q-1} \circ \ldots \circ h_1 \circ h_0$$

Theorem 4.2. Under the compositional smoothness setting (4.1), assume condition (M1,M2) and denote $\kappa^- = \inf_{\boldsymbol{x}\in[0,1]^d} K(\boldsymbol{x}), \kappa^+ = \sup_{\boldsymbol{x}\in[0,1]^d} K(\boldsymbol{x})$. Let $\mathcal{F}_n^*$ be a ReLU DNN family with proper architectures and size constraint $L_n^* \asymp \log(n)$,

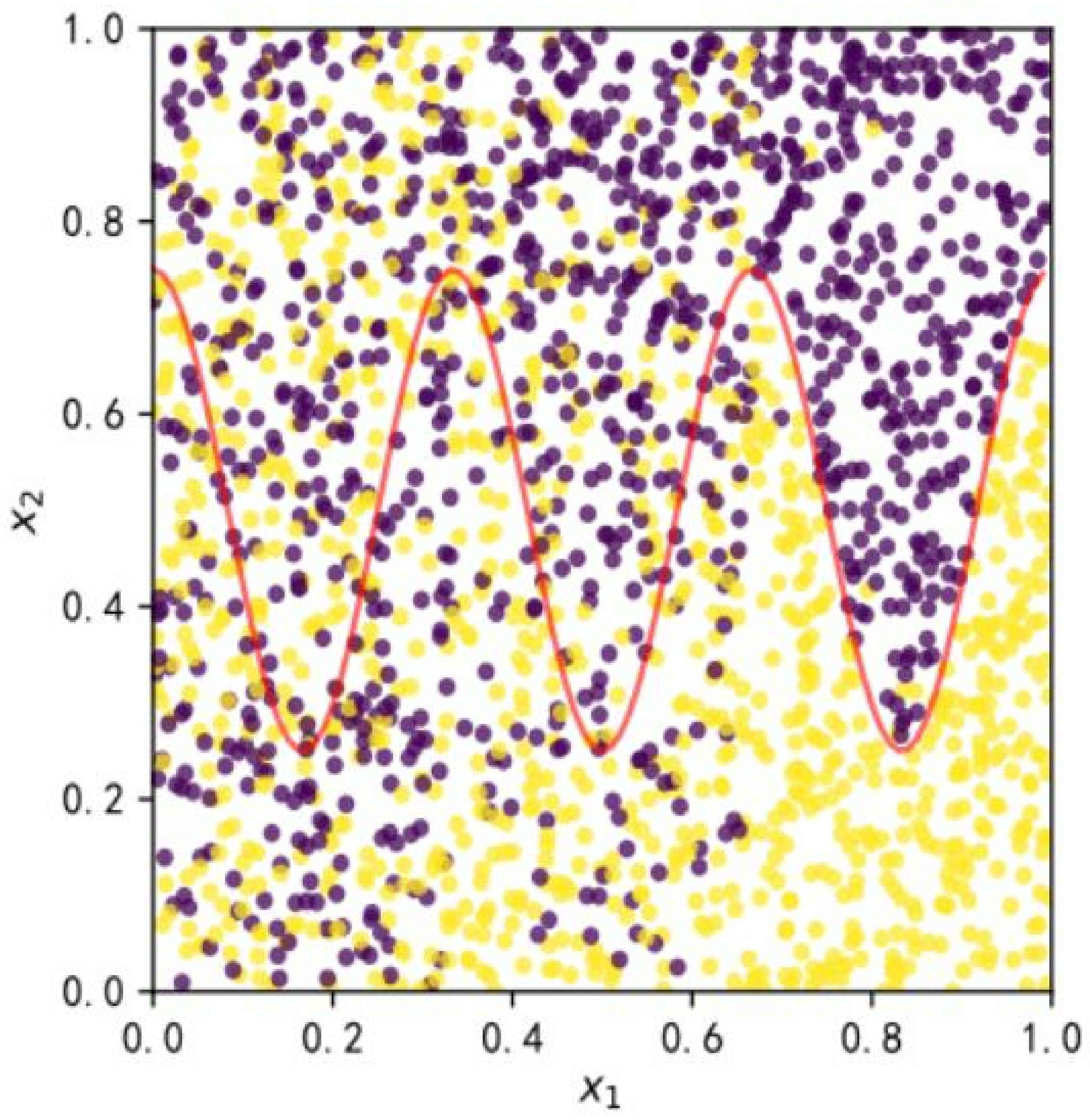
$$N_n^* \asymp n^{\frac{\kappa^+ d^*}{(\kappa^+ + 2)\beta^* + d^* \kappa^+}} \log^{d-1}(n), \quad S_n^* \asymp n^{\frac{\kappa^+ d^*}{(\kappa^+ + 2)\beta^* + d^* \kappa^+}} \log^{d}(n).$$

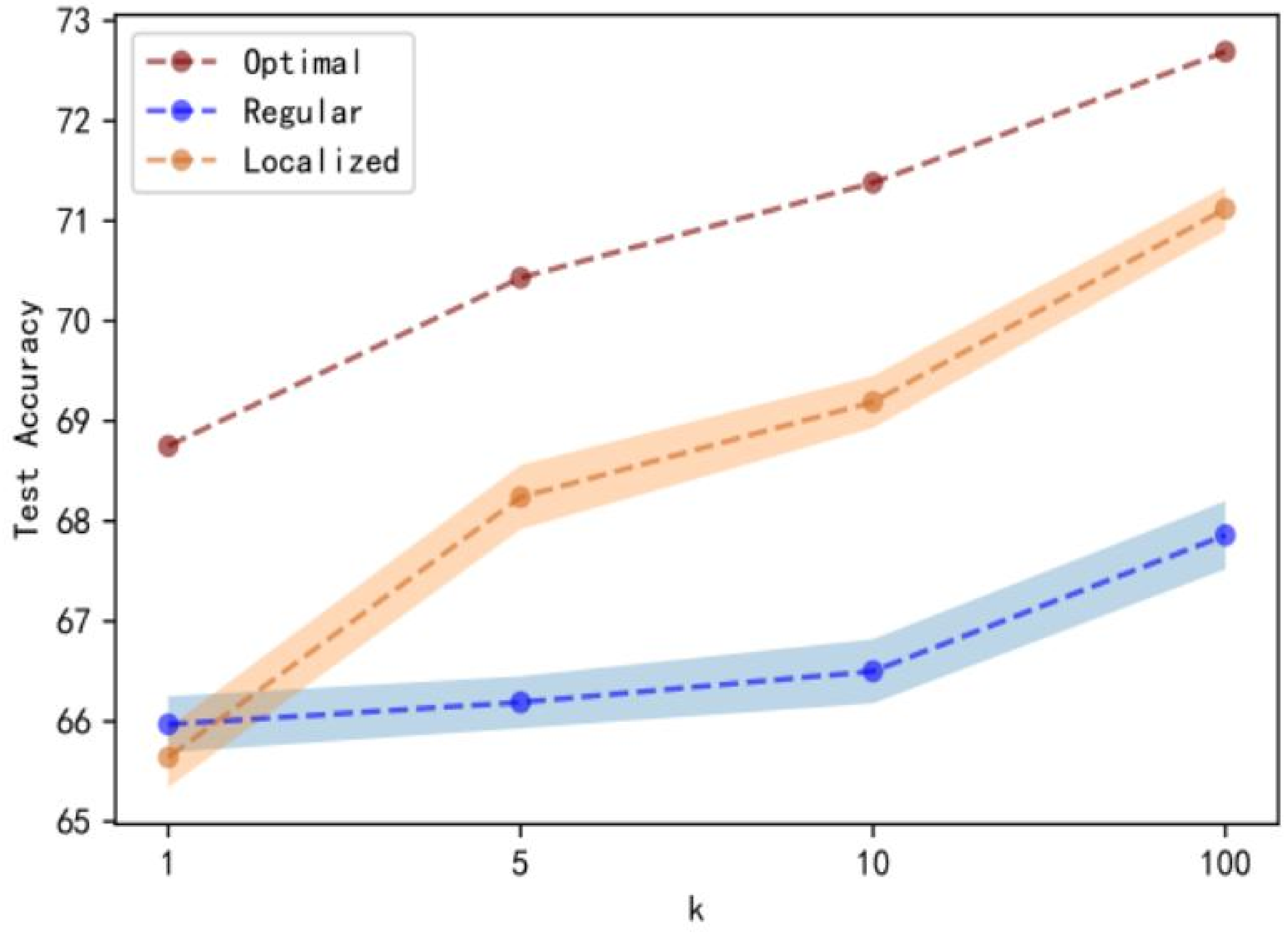
Then, with probability $\xrightarrow{n\to\infty}$ 1, the empirical 0-1 loss minimizer within $\mathcal{F}_n$ satisfies

$$\inf_{\widehat{f}_n \in \mathcal{F}_n^*} \sup_{C^* \in \mathcal{C}(d^*, \beta^*)} \mathbb{E}(R(\widehat{f}_n) - R(C^*)) = \widetilde{O}\left(n^{-\frac{(\kappa^- + 1)\beta^*}{(\kappa^- + 2)\beta^* + \kappa^- d^*}}\right).$$

CAD-NN: Simulation

Recall the 2D example, where k governs the inconsistency.





CAD-NN: Boundary Complexity

In learning theory, the model complexity (how large is the model) is of critical importance, especially for model generalization.

Bag of Tricks $\xrightarrow{\text{regularize}}$ Model Complexity $\xrightarrow{\text{ensure}}$ Good Generalization

However, for classification, existing regularizations may be insufficient or irrelevant.

In classification, to achieve good generalization, the complexity to control:

Classifier $\{f(x): f \in \mathcal{F}\}$ ✗

Boundary Complexity $\{\{x: f(x) \geq 0\}: f \in \mathcal{F}\}$ ✓

- The boundary complexity measurement is far less explored, classical notions e.g., covering number, may be inadequate for deep learning.
- Given a boundary complexity, regularizing it during neural network training can be challenging. Adversarial training can be thought of as a regularization for boundary complexity

CAD-NN: Boundary Complexity

A (proof-of-concept) step towards this underexplored direction

Consider ReLU neural network, where the decision boundary is **piecewise linear**!

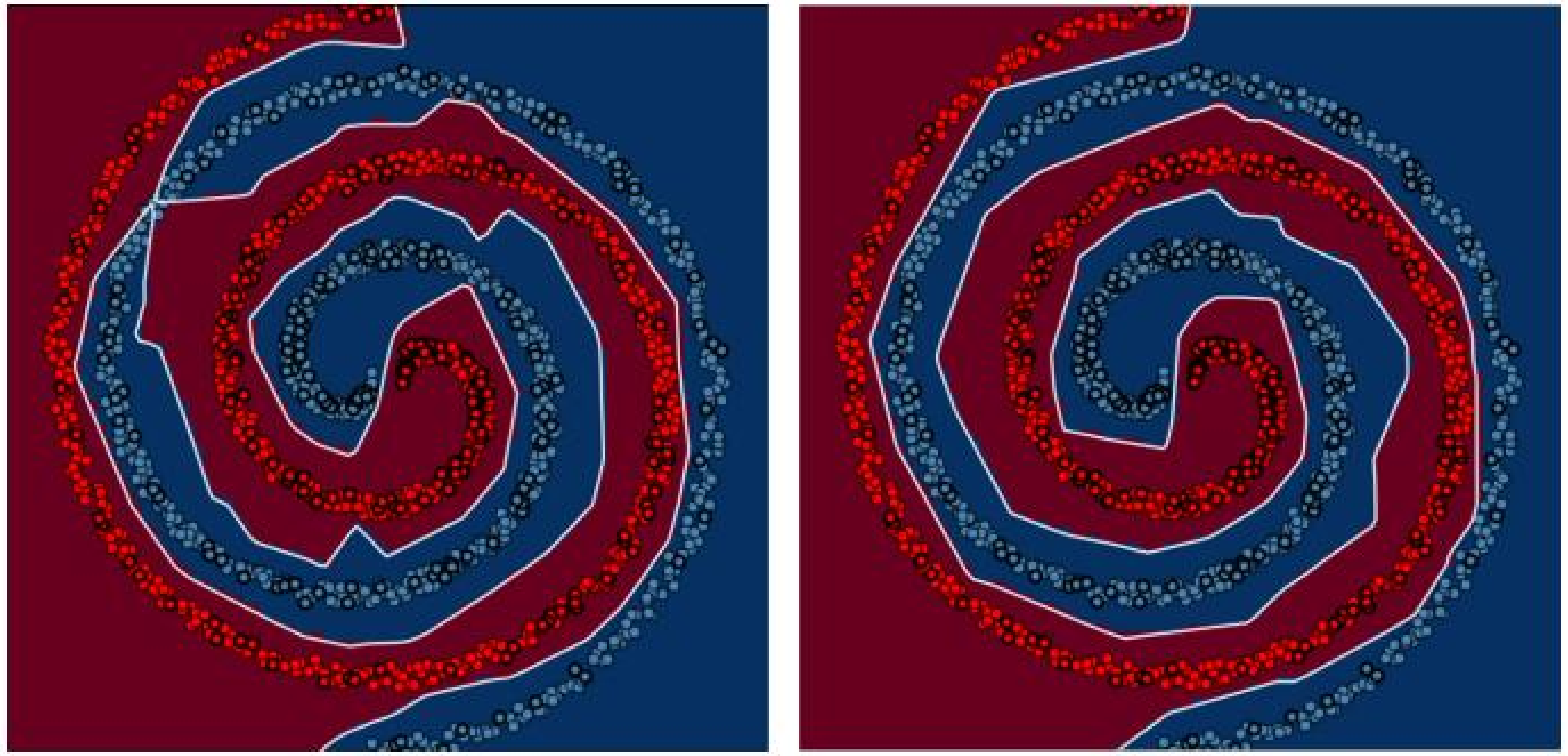
Boundary complexity can be conveniently characterized by the **#Boundary (number of linear pieces)**.

Pros:

- Well-defined
- Intuitive
- Synergy with #Total (total number of linear pieces)

Cons:

- Over-simplified
- Not easy to calculate

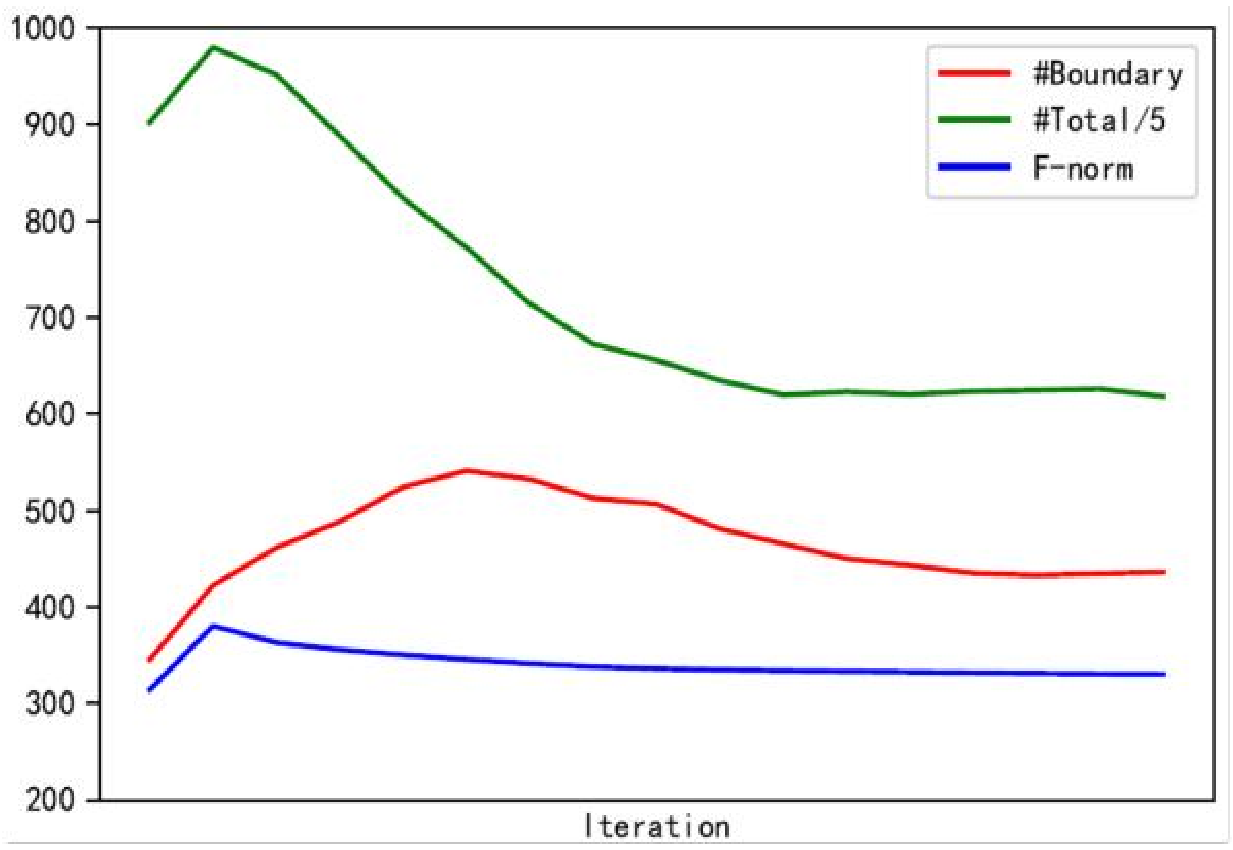


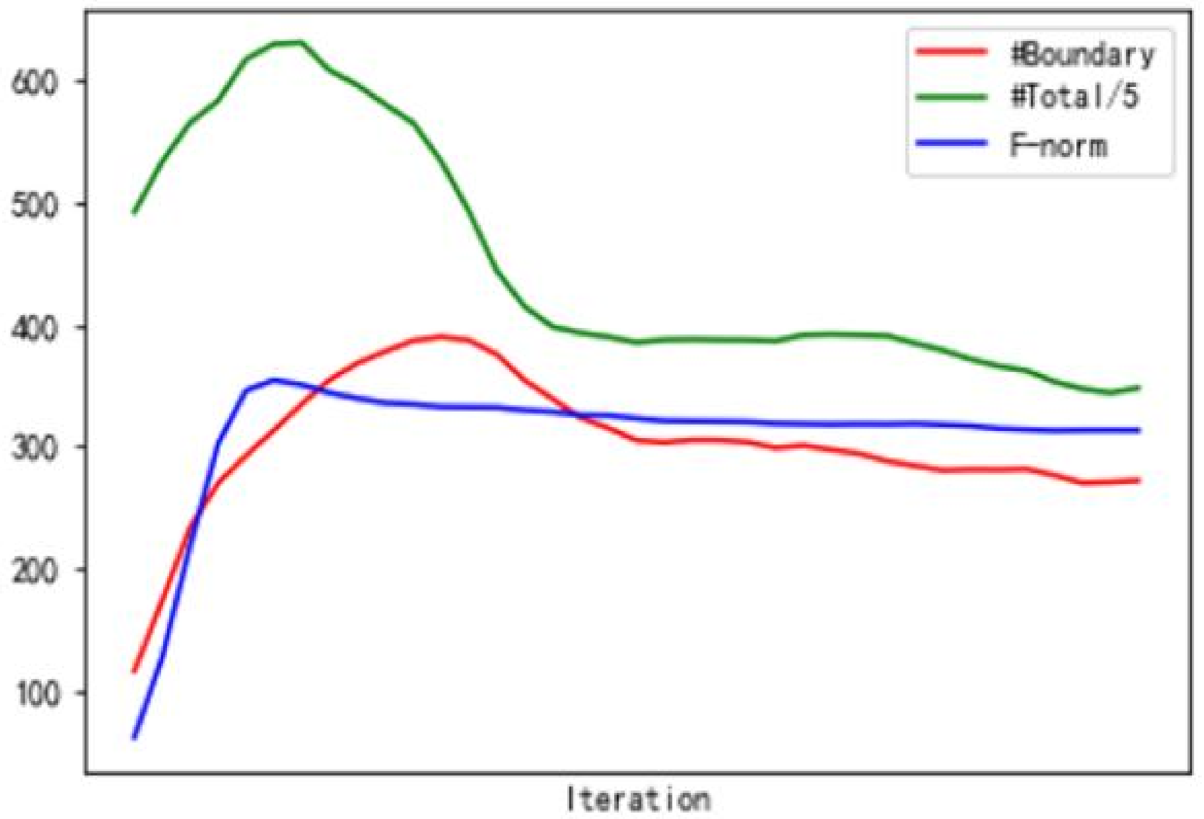
CAD-NN: Boundary Complexity

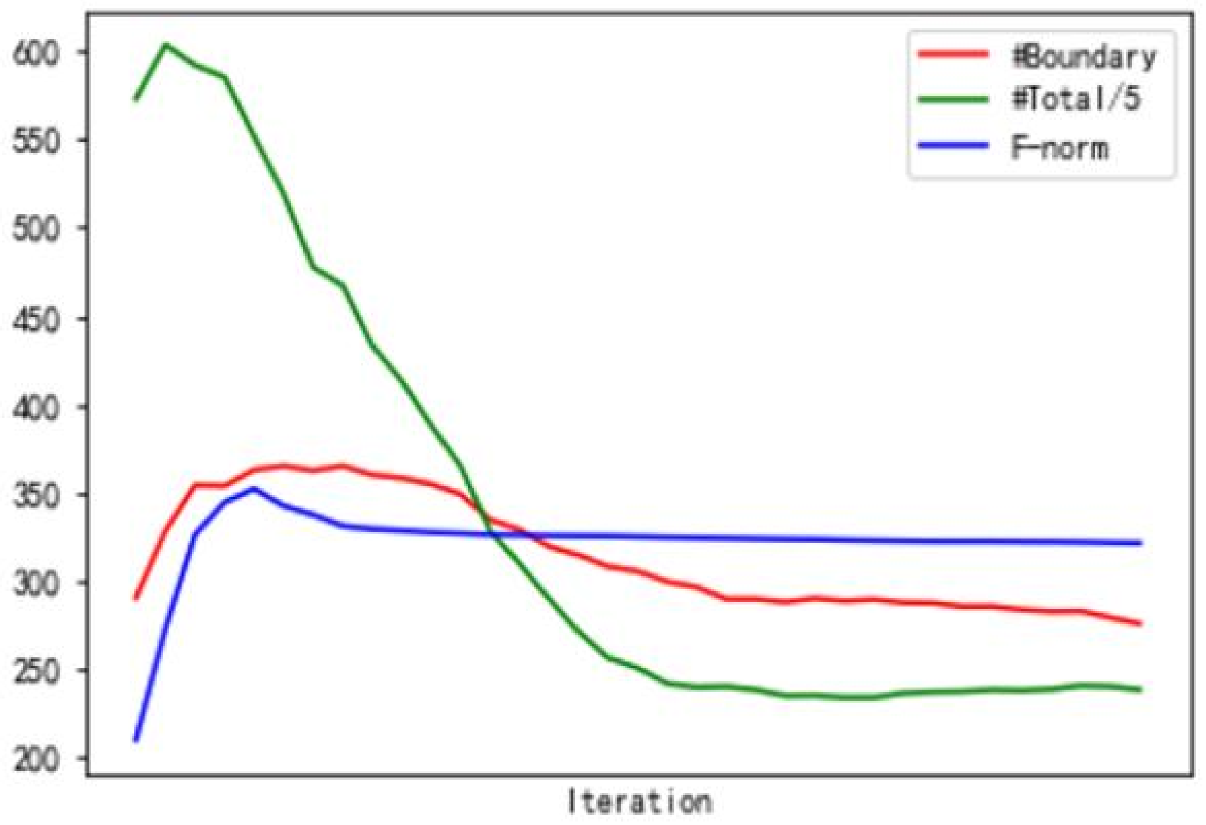
A (proof-of-concept) step towards this underexplored direction: We propose a method to explicitly count the number of boundary pieces, with the help of Tropical Geometry.

Two take home messages:

- **Boundary complexity is different from functional complexity**



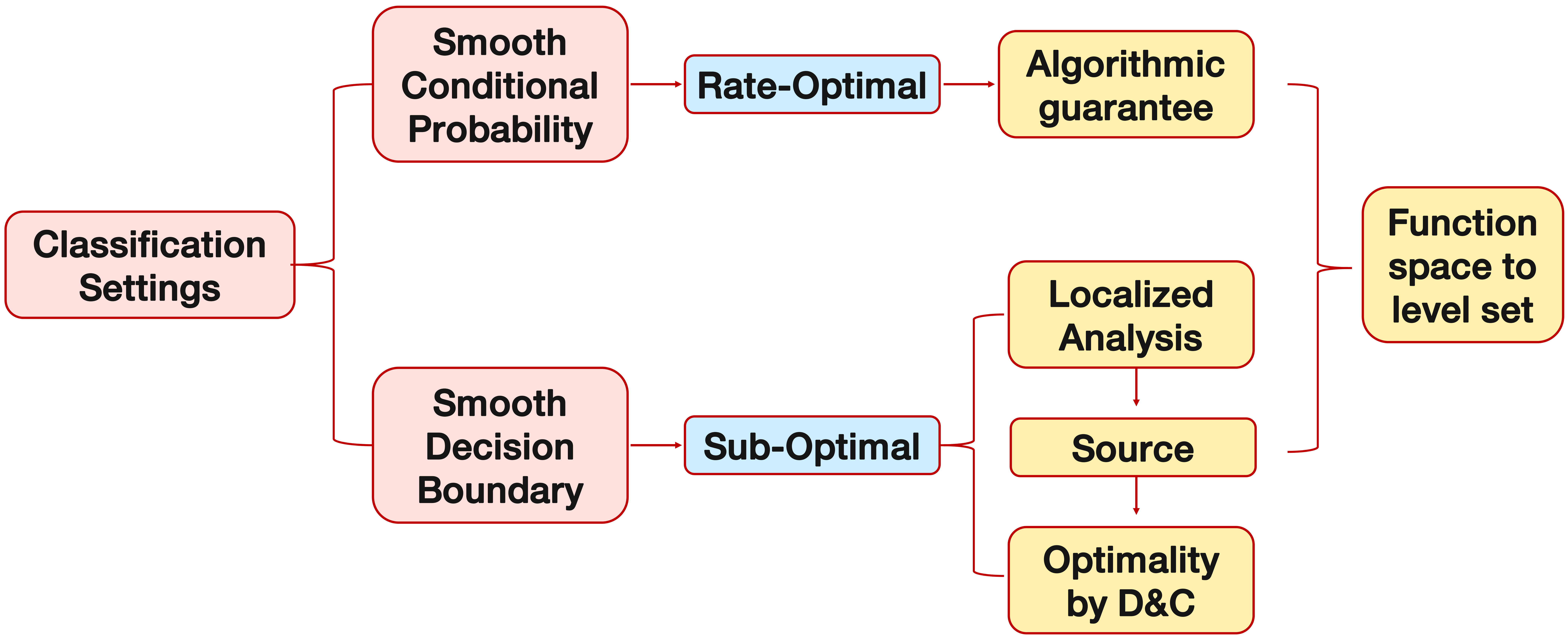




- **Boundary complexity can have negative correlation with Classification Robustness**

	#Boundary	#Total	F-norm	Acc%	$R(0.02)$
Initial	90 (61)	2432 (179)	20 (0.71)	50.2 (1.2)	-
CE	377 (31)	1915 (207)	283 (11)	93.60 (1.8)	94.3 (2.2)
Noisy	272 (33)	1493 (114)	322 (17)	99.15 (0.56)	98.1 (0.51)
Adv	259 (21)	1241 (135)	356 (19)	99.35 (0.38)	98.9 (0.36)

Summary



Statistics has a lot more to offer for theoretical understanding of deep learning.