Calibration in Deep Learning: Theory & Practice

Preetum Nakkiran

Aspen Center for Physics | Apple Inc. | Feb 28 2023

Motivation

"Why do we get more than we asked for in Deep Learning?"

Small Train Loss

SGD

- Small Test Loss
- "Good representations"
- Good transfer & fine-tuning
- Good OOD robustness
- **Good calibration**

Motivation

"Why do we get more than we asked for in Deep Learning?"

Small Train Loss → SGD →

- Small Test Loss
- "Good representations"
- Good transfer & fine-tuning
- Good OOD robustness
- **Good calibration**

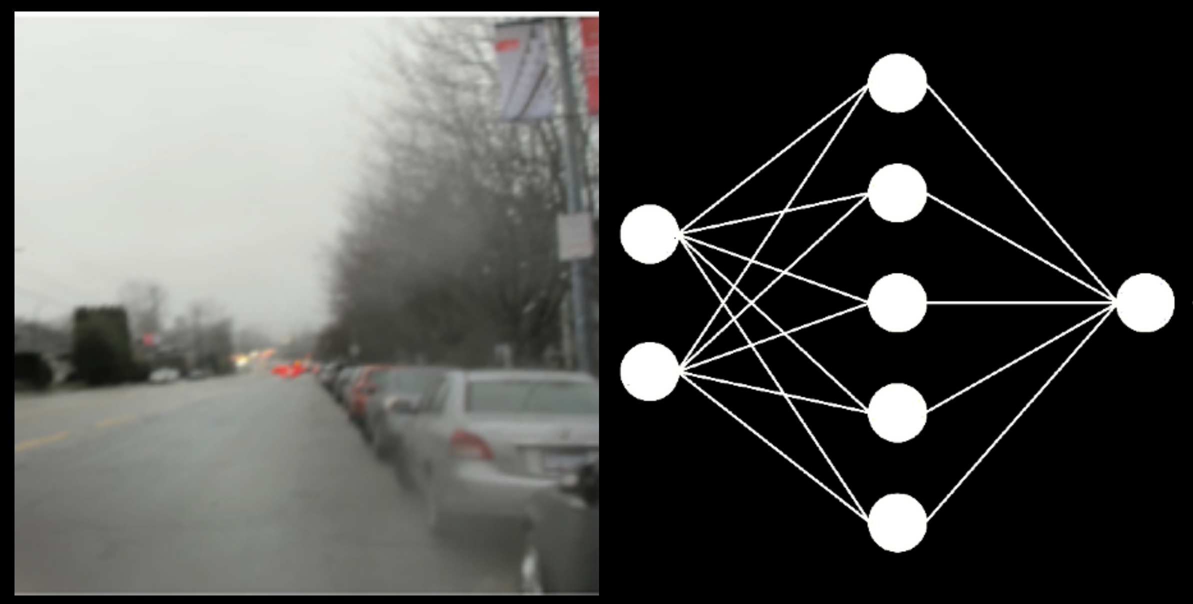
Goal: What's important about this box?

What is Calibration?

Setting: Binary classification

Test distribution D over $(x, y) \in \mathcal{X} \times \{0,1\}$

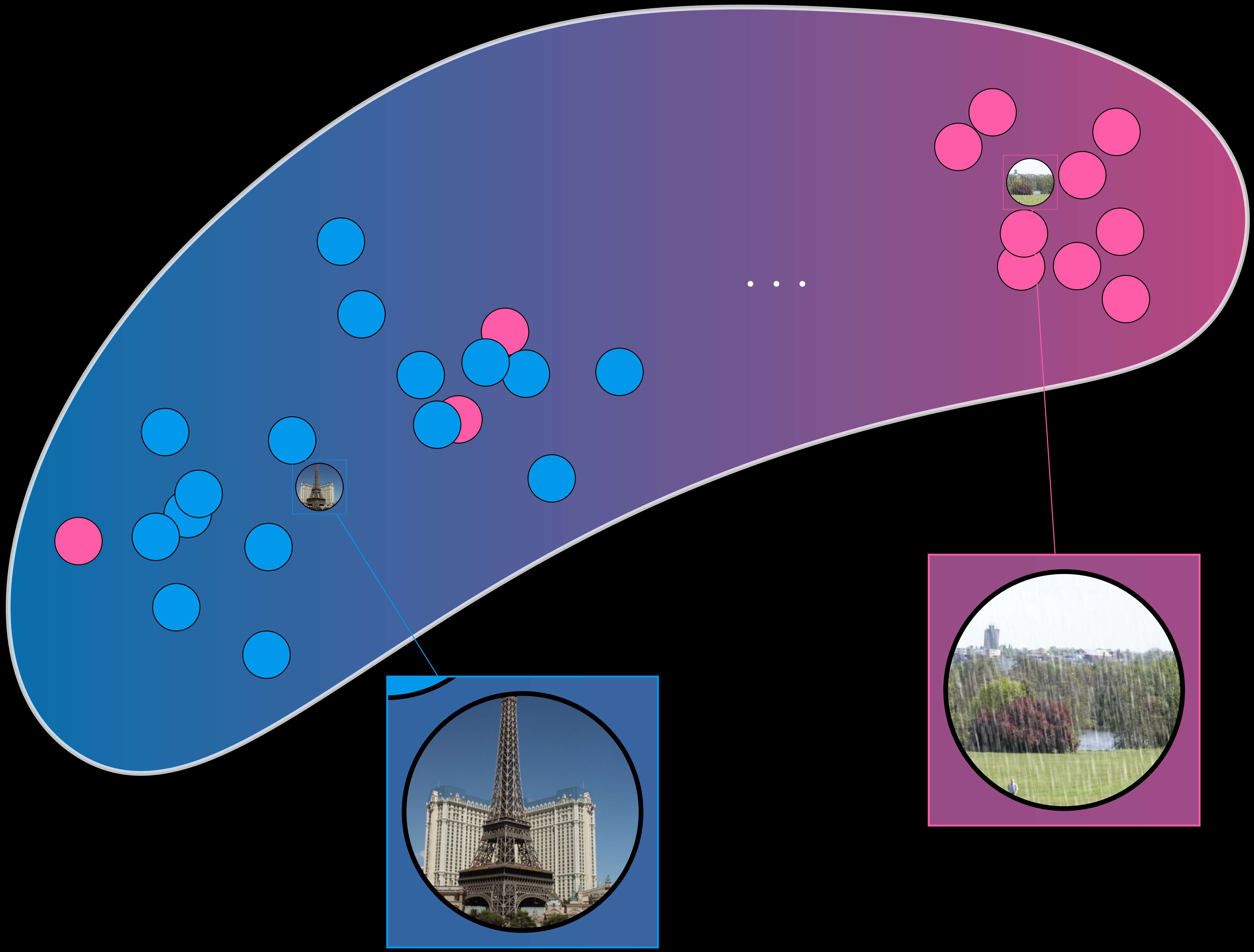
Predictor $f : \mathcal{X} \rightarrow [0,1]$ *"f(x) is confidence of y(x) = 1"*



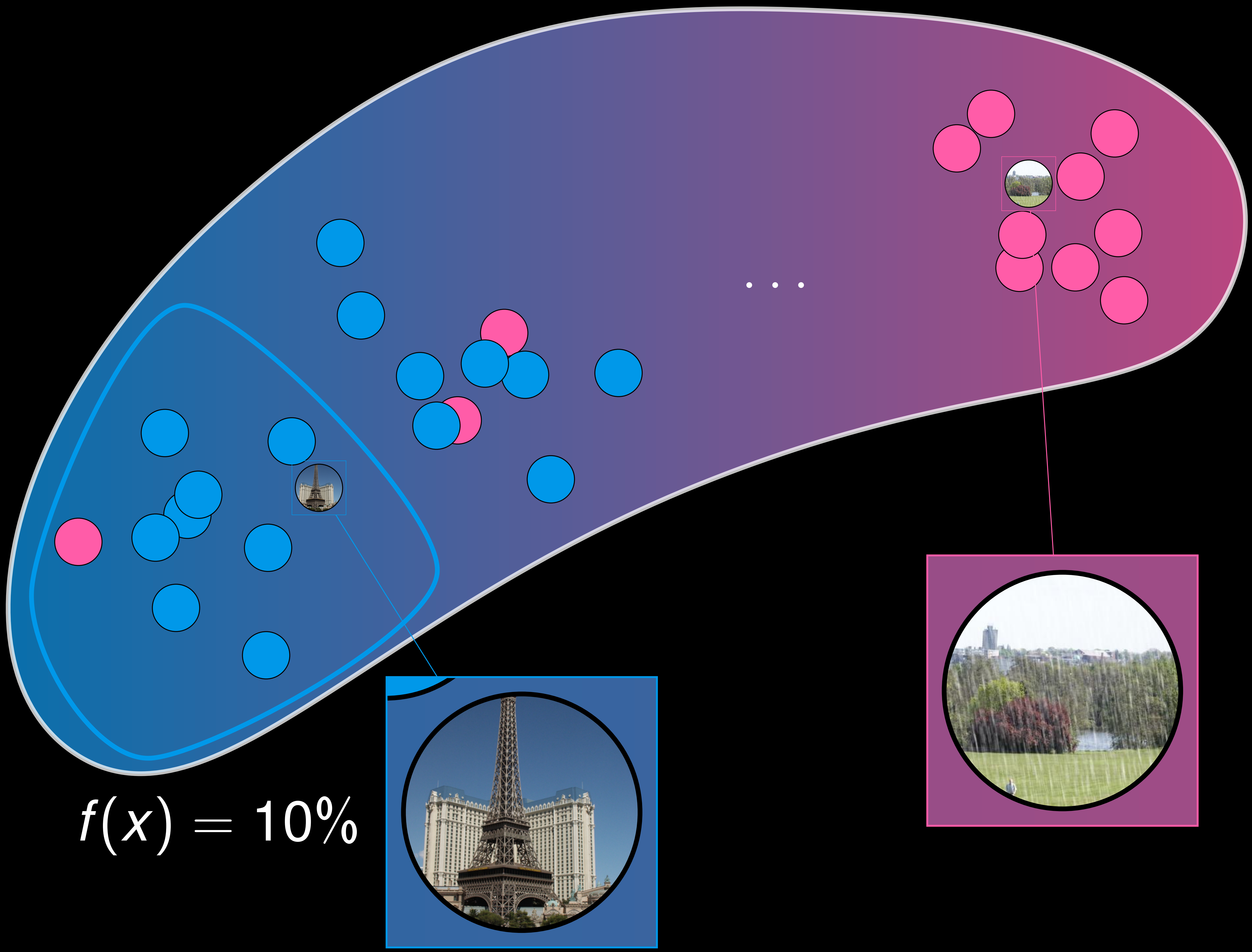
$\hat{p}(\text{rain}) \in [0,1]$

Perfect calibration is a property of the pair (f, D)

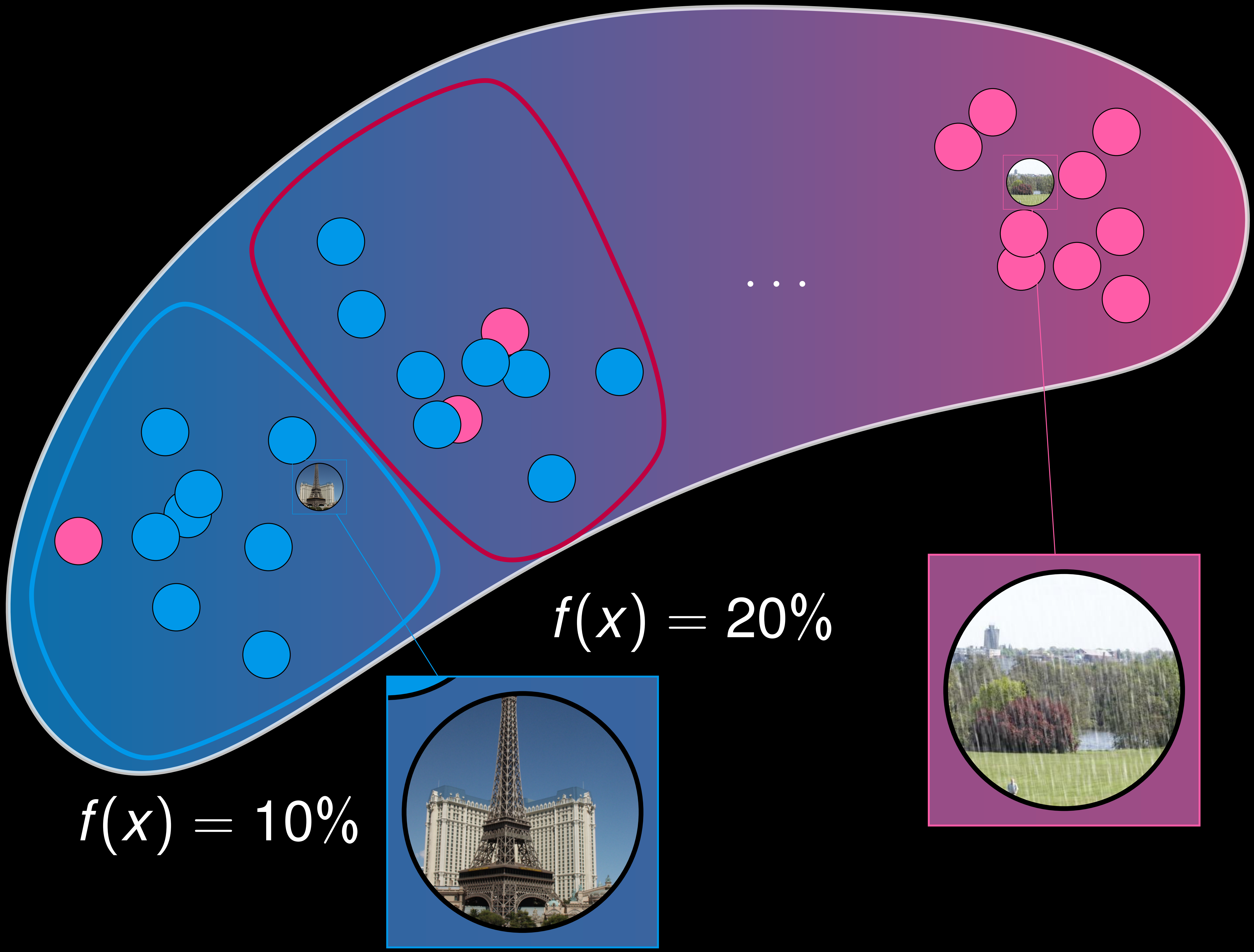
Calibration: Predictor f is perfectly calibrated w.r.t. distribution D if...



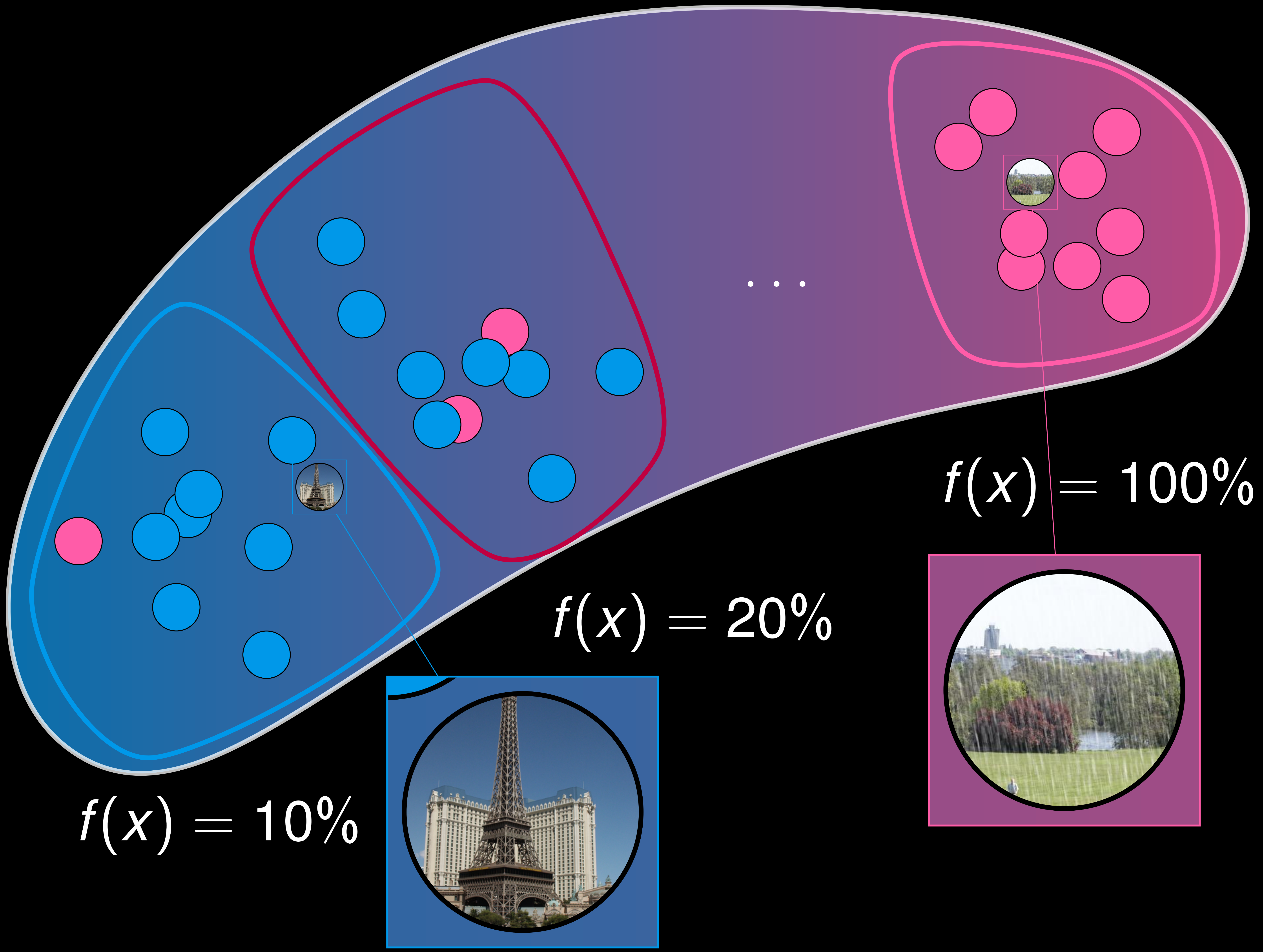
Calibration: Predictor f is perfectly calibrated w.r.t. distribution D if...



Calibration: Predictor f is perfectly calibrated w.r.t. distribution D if...



Calibration: Predictor f is perfectly calibrated w.r.t. distribution D if...



Perfect Calibration:

Predictor f is perfectly calibrated w.r.t. D if

$$\forall \ell \in [0,1] : \quad \mathbb{E}_{x,y\sim D}[y \mid f(x) = \ell] = \ell$$

Perfect Calibration:

Predictor f is perfectly calibrated w.r.t. D if

$$\forall \ell \in [0,1] : \quad \mathbb{E}_{x,y \sim D}[y \mid f(x) = \ell] = \ell$$

What's calibration good for?

1. **Interpretability**: $f(x)$ is a meaningful quantity, "confidence that $y = 1$"
   e.g. doctor informing patient of "80% probability of heart disease"

2. **Operational Uncertainty:** Systems downstream of $f(x)$ can behave differently on "high confidence" vs. "low confidence" inputs

$$\Pr[y=1 \mid f(x) = 0.5] = 0.5$$
$$\Pr[y=1 \mid f(x) = 0.8] = 0.8$$
$$\Pr[y=1 \mid f(x) = 1.0] = 1.0$$

<u>Perfect Calibration</u>:

Predictor f is perfectly calibrated w.r.t. D if

$$\forall \ell \in [0,1] : \quad \mathbb{E}_{x,y\sim D}[y \mid f(x) = \ell] = \ell$$

What's calibration good for?

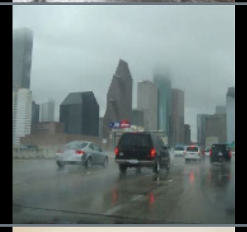
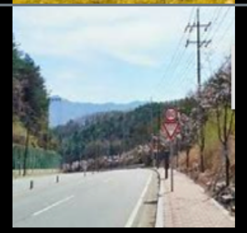
1. **Interpretability**: $f(x)$ is a meaningful quantity, "confidence that $y = 1$"
   e.g. doctor informing patient of "80% probability of heart disease"

2. **Operational Uncertainty:** Systems downstream of $f(x)$ can behave differently on "high confidence" vs. "low confidence" inputs

   Pr[y=1 | f(x) = 0.5] = 0.5
   Pr[y=1 | f(x) = 0.8] = 0.8
   Pr[y=1 | f(x) = 1.0] = 1.0

3. **Interesting:** "Models knows when it doesn't know" self-consistency

Calibration is **orthogonal** to accuracy

input x	ground-truth y	prediction f(x)
	1	0.5
	1	0.5
	1	0.5
	0	0.5
	0	0.5
	0	0.5

Goal

Understand <u>when</u> and <u>why</u> (DL) models are well-calibrated

(& what factors affect calibration)

This Talk

1. How to Define & Measure Miscalibration

“A Unifying Theory of Distance from Calibration.” *STOC 2023*. *arXiv:2211.16886*. [Błasiok, Gopalan, Hu, **N.**]

2. Empirical Conjectures

“The Calibration Generalization Gap.” *arXiv:2210.01964*. [Carrell, Mallinar, Lucas, **N.**]
+ upcoming work

3. Theory

“When Loss Minimization Yields Calibration.” *In preparation*. [Błasiok, Gopalan, Hu, **N.**]

“Loss minimization yields multicalibration for large neural networks.” *In submission*. [Kalai]

This Talk

1. How to Define & … e Miscalibration

SKIP

“A Unifying T … Distance from Calibration.” *STOC 2023. arXiv:2211.16886*. [Błasiok, Gopalan, Hu, **N.**]

2. Empirical Conjectures

“The Calibration Generalization Gap.” *arXiv:2210.01964*. [Carrell, Mallinar, Lucas, **N.**]
+ upcoming work

3. Theory

“When Loss Minimization Yields Calibration.” *In preparation*. [Błasiok, Gopalan, Hu, **N.**]

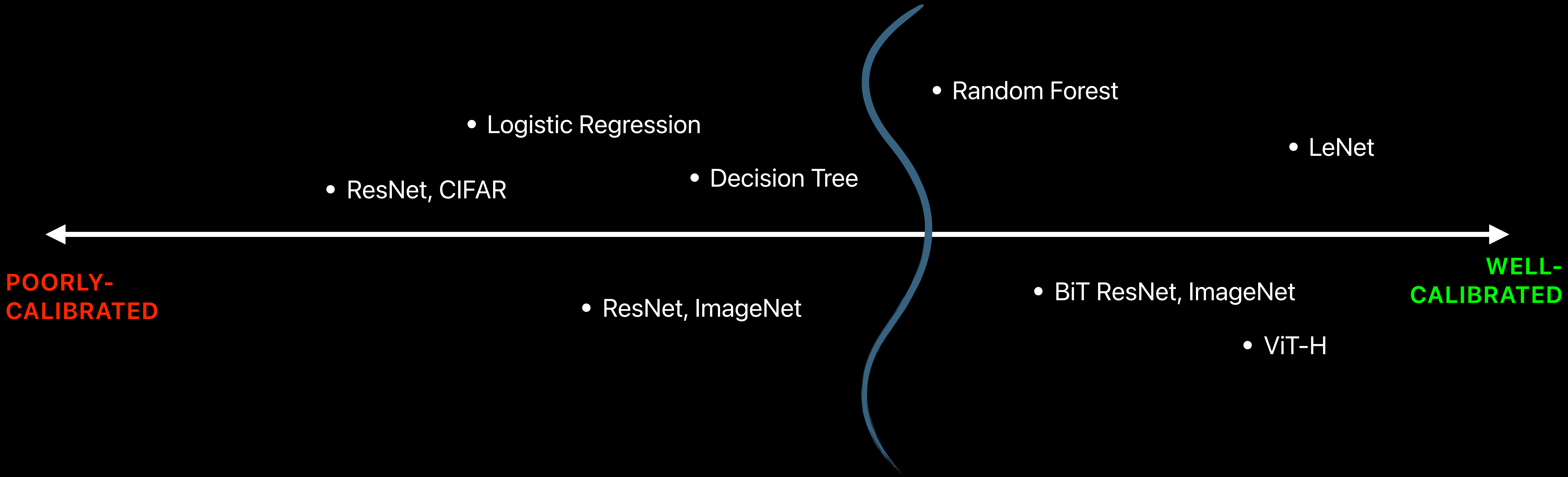
“Loss minimization yields multicalibration for large neural networks.” *In submission*. [Kalai]

Part 2.
Calibration of DNNs, Experimentally

Can we empirically characterize which DNNs have small calibration error?

Landscape of Calibration

- {Model, Data-distribution}

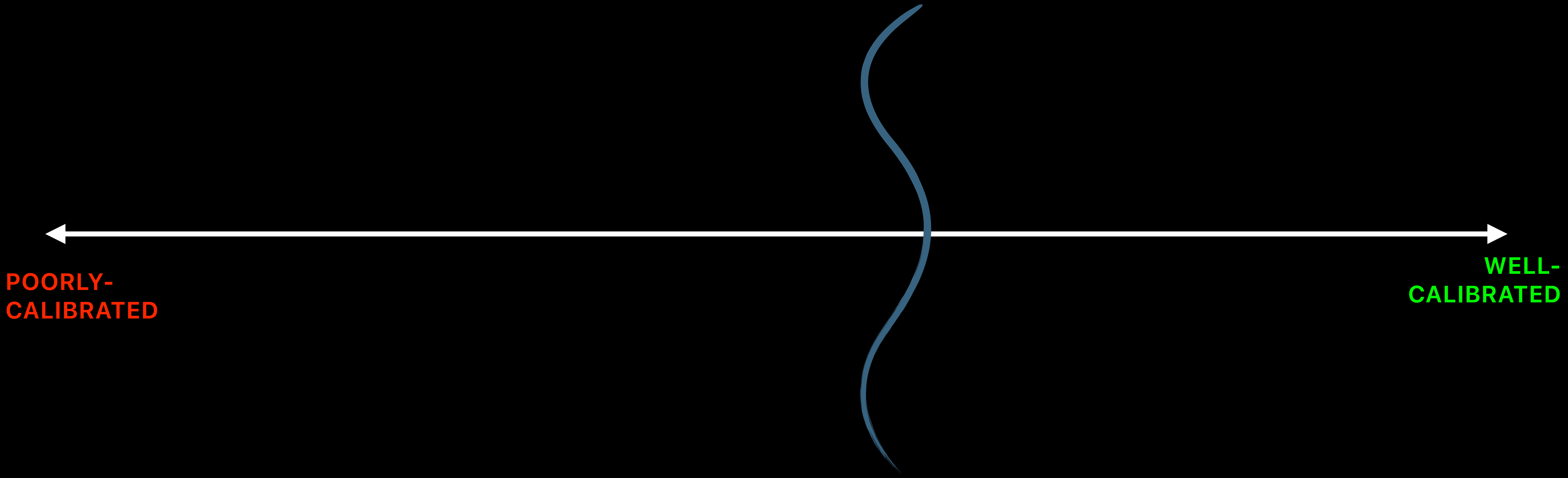


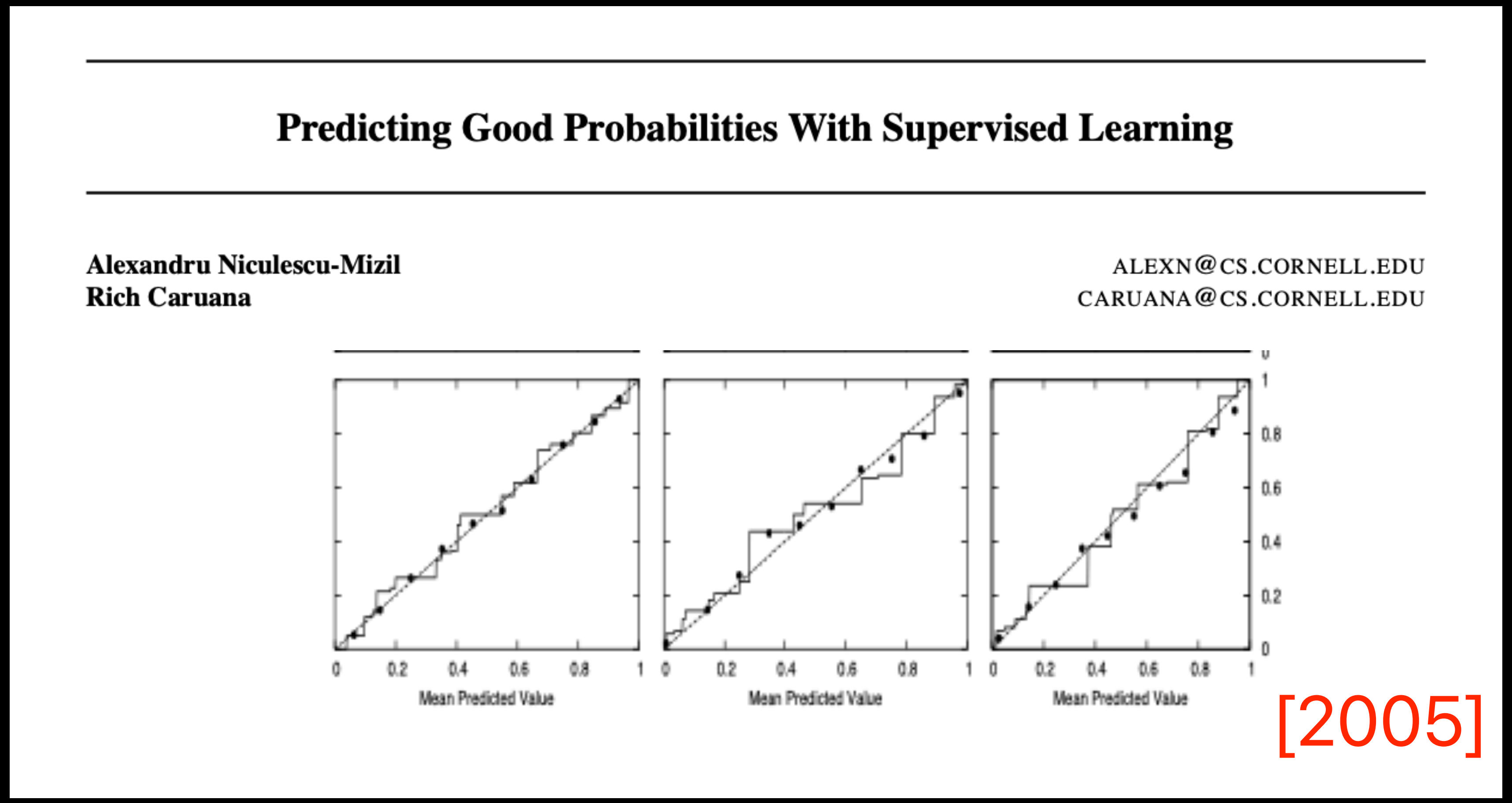
Landscape of Calibration

No single design choice determines calibration:

- Not just architecture
- Not just model size
- Not just test accuracy/loss

    ex: small 3-layer MLP is well-calibrated on ImageNet


POORLY-
CALIBRATED
WELL-
CALIBRATED



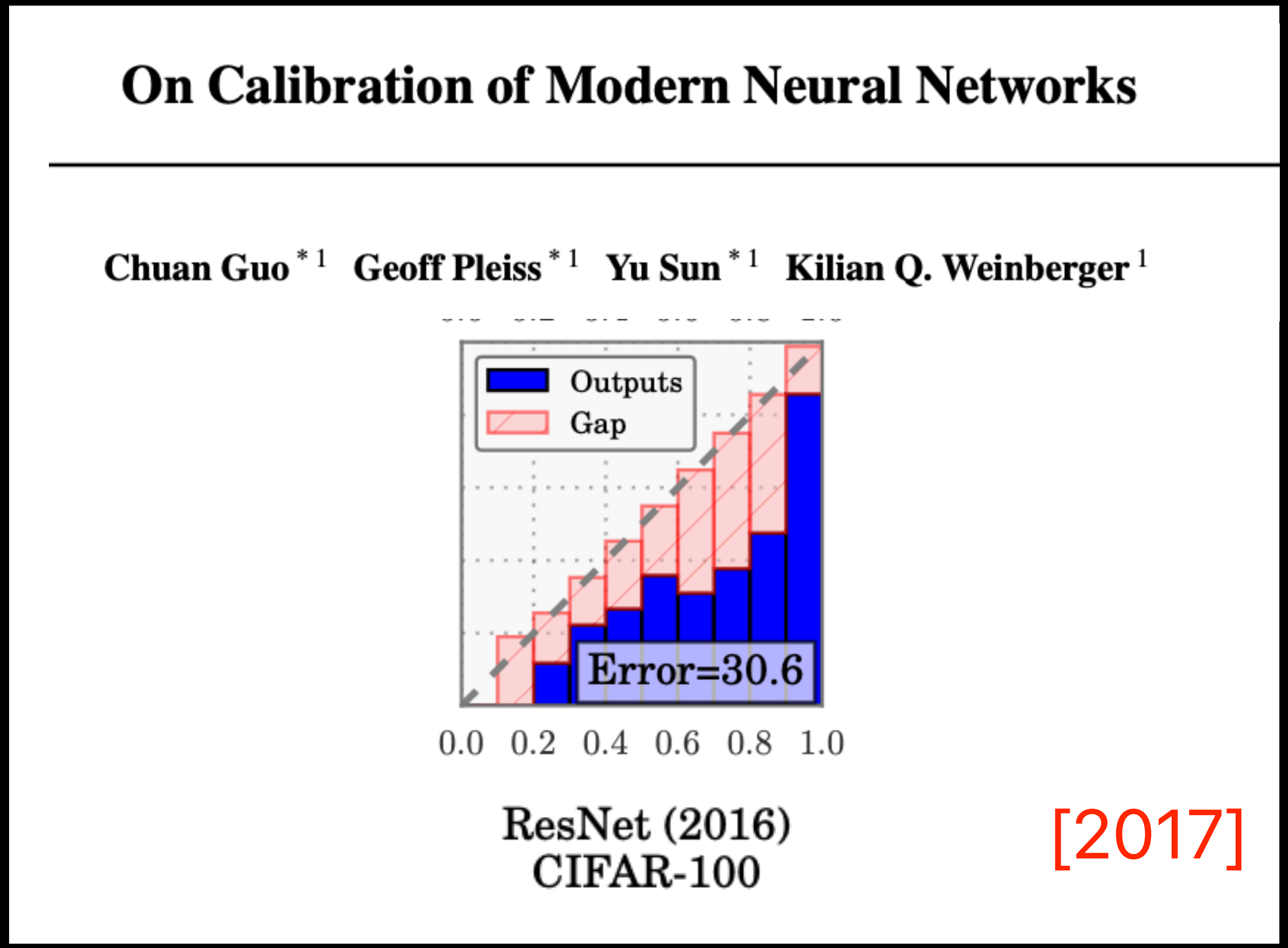
Predicting Good Probabilities With Supervised Learning

Alexandru Niculescu-Mizil ALEXN@CS.CORNELL.EDU
Rich Caruana CARUANA@CS.CORNELL.EDU

[2005]

POORLY-
CALIBRATED

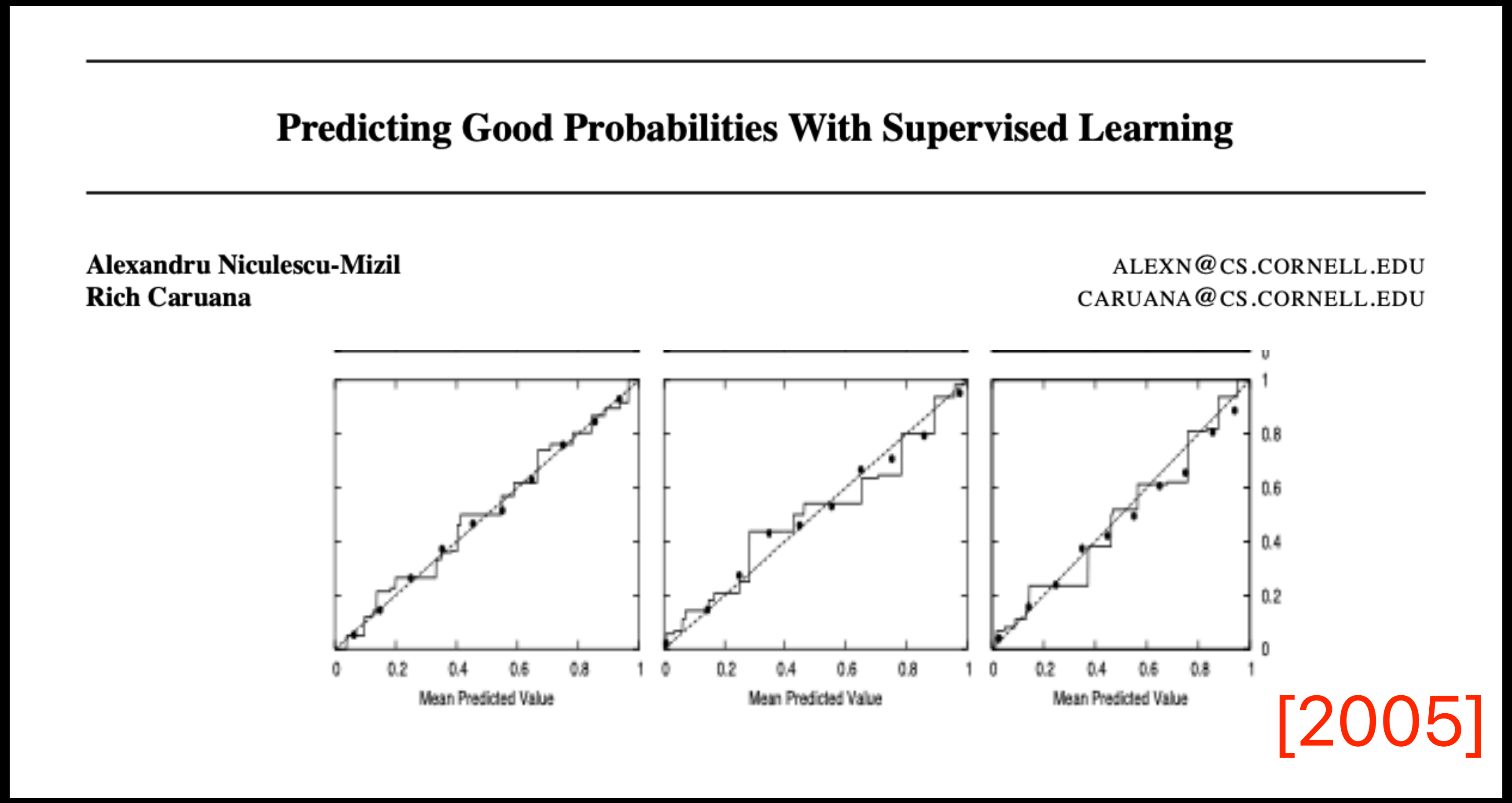
WELL-
CALIBRATED


On Calibration of Modern Neural Networks

Chuan Guo [*1] Geoff Pleiss [*1] Yu Sun [*1] Kilian Q. Weinberger [1]

ResNet (2016)
CIFAR-100

[2017]

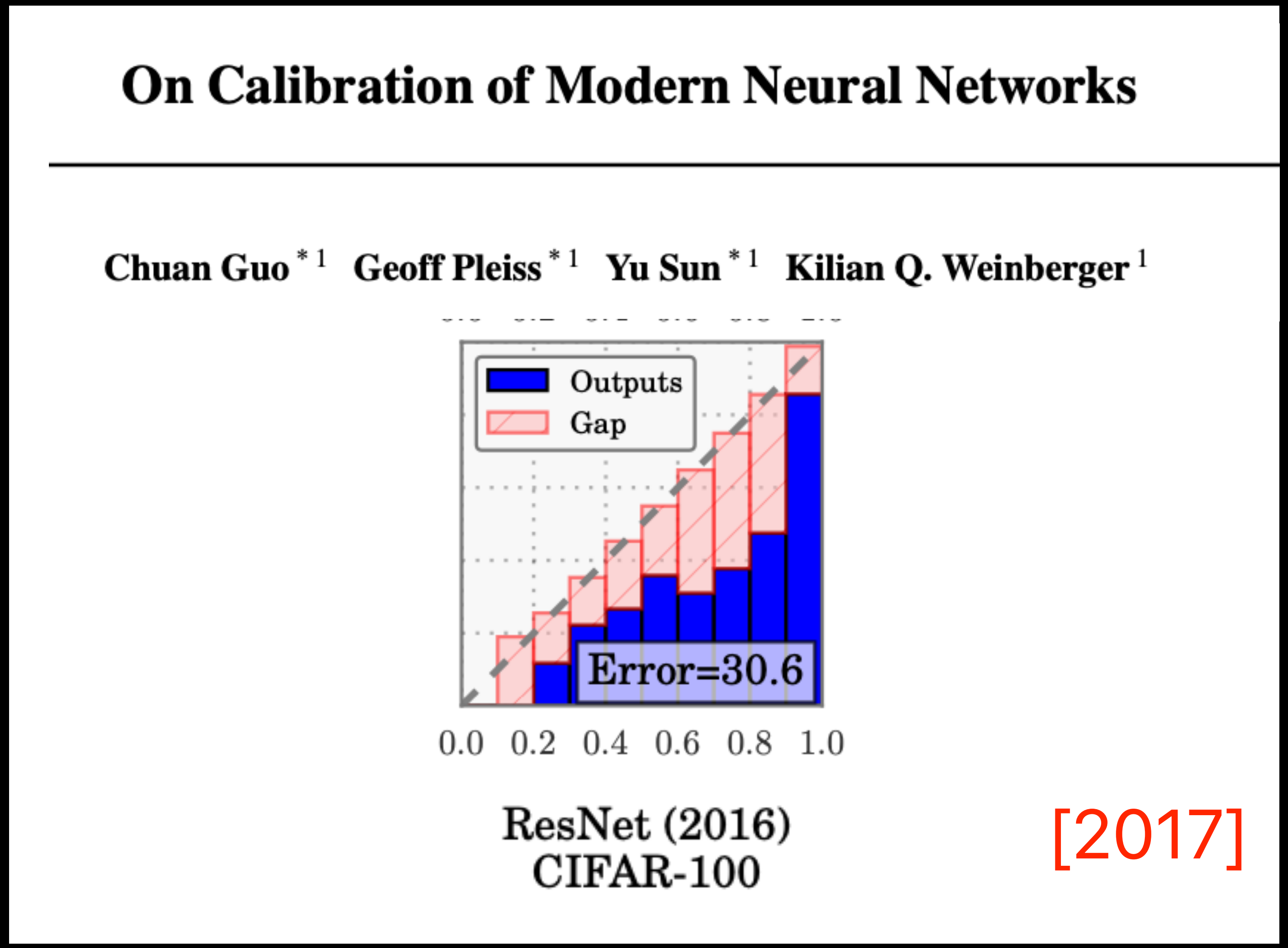

Predicting Good Probabilities With Supervised Learning

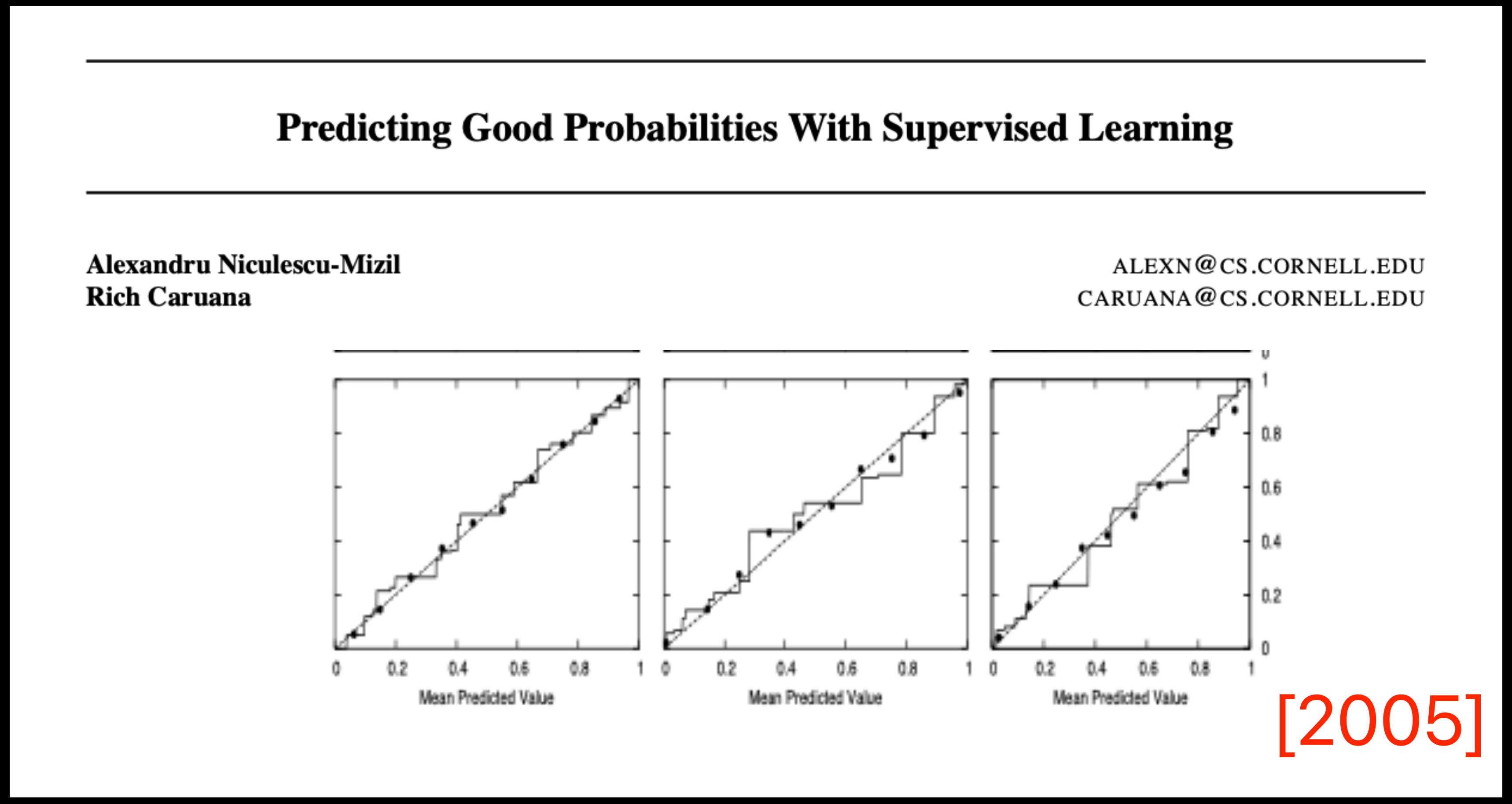
Alexandru Niculescu-Mizil ALEXN@CS.CORNELL.EDU
Rich Caruana CARUANA@CS.CORNELL.EDU

[2005]

POORLY-
CALIBRATED

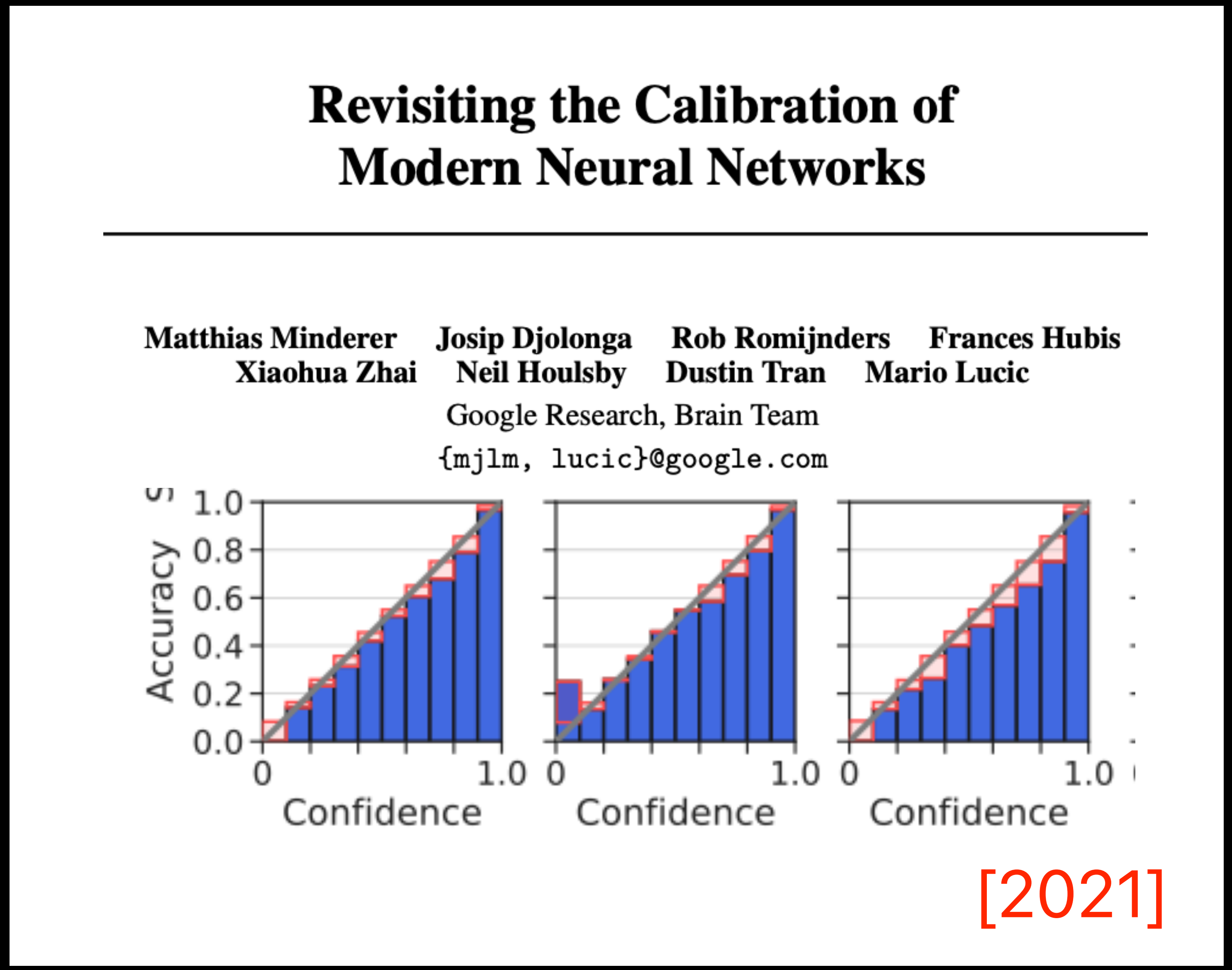
WELL-
CALIBRATED





POORLY-CALIBRATED

WELL-CALIBRATED



Proposal: Study test-calibration as we study test-error.

Fundamental Decomposition

$$\underbrace{\mu\text{Test}}_{\text{Calibration Error on Test Set}} \leq \boxed{\underbrace{\mu\text{Train}}_{\text{Calibration Error on Train Set}}} + \boxed{\underbrace{|\mu\text{Test} - \mu\text{Train}|}_{\text{Calibration Generalization Gap}}}$$

Proposal: Study test-calibration as we study test-error.

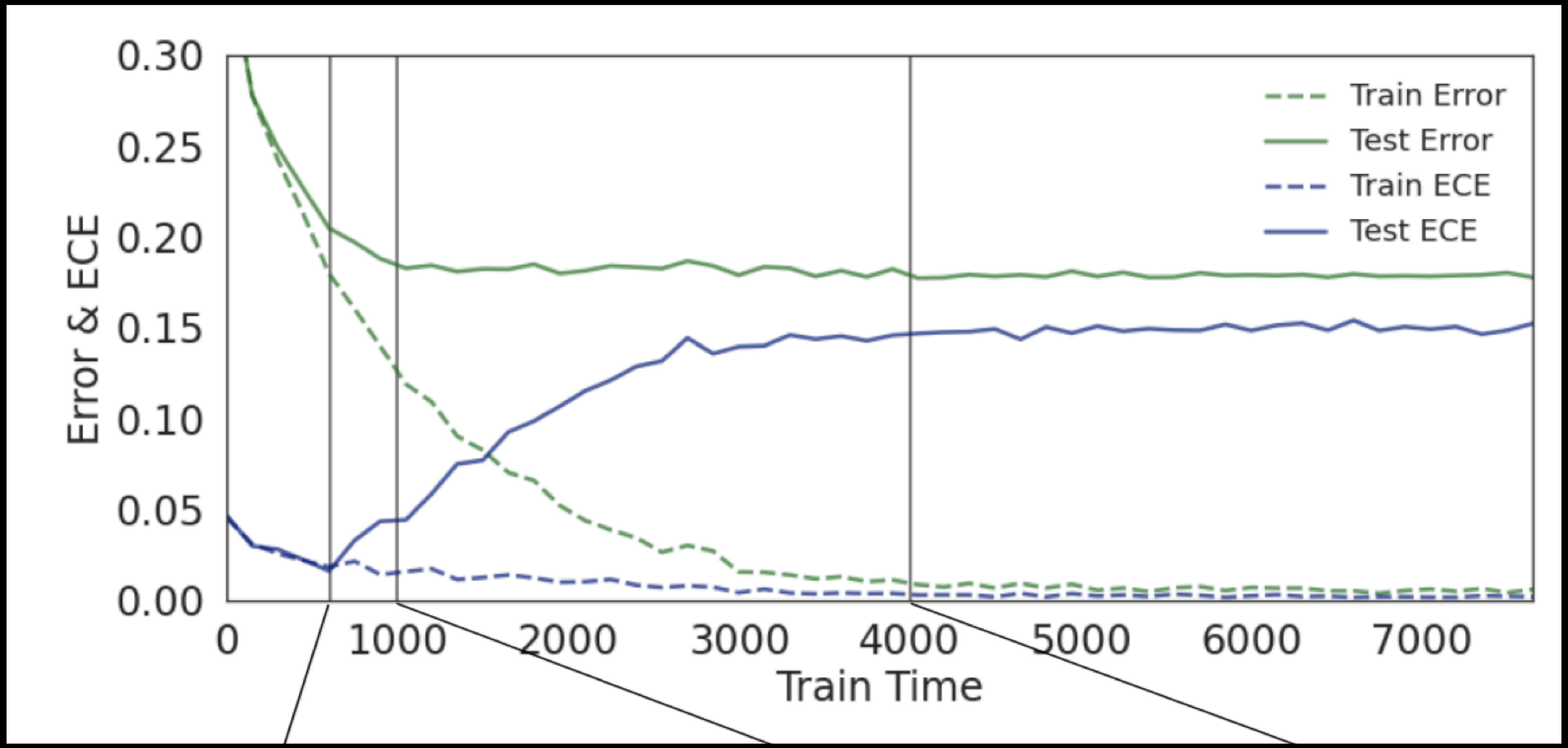
Fundamental Decomposition

$$\underbrace{\mu\text{Test}}_{\text{Calibration Error on Test Set}} \leq \boxed{\underbrace{\mu\text{Train}}_{\text{Calibration Error on Train Set}}} + \boxed{\underbrace{|\mu\text{Test} - \mu\text{Train}|}_{\text{Calibration Generalization Gap}}}$$

Trivial, BUT:

1. Suggests methodology: study each part separately
2. Insightful: parts are simpler than the whole

(higher=worse)



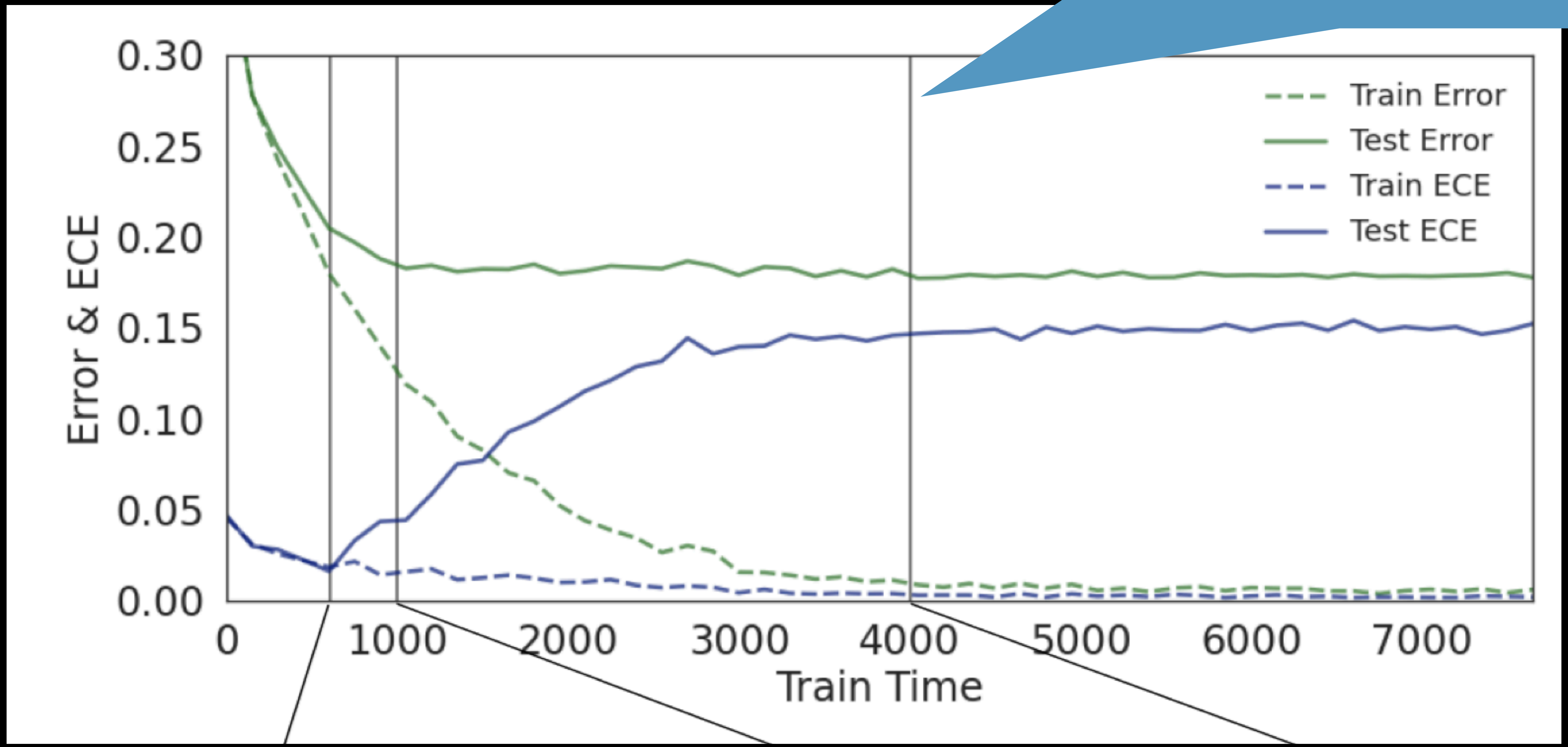
ViT on binary-CIFAR-10

End of training:
models are overconfident

Train {Error,CE} ≈ 0

Test {Error,CE} $\gg 0$

(higher=worse)



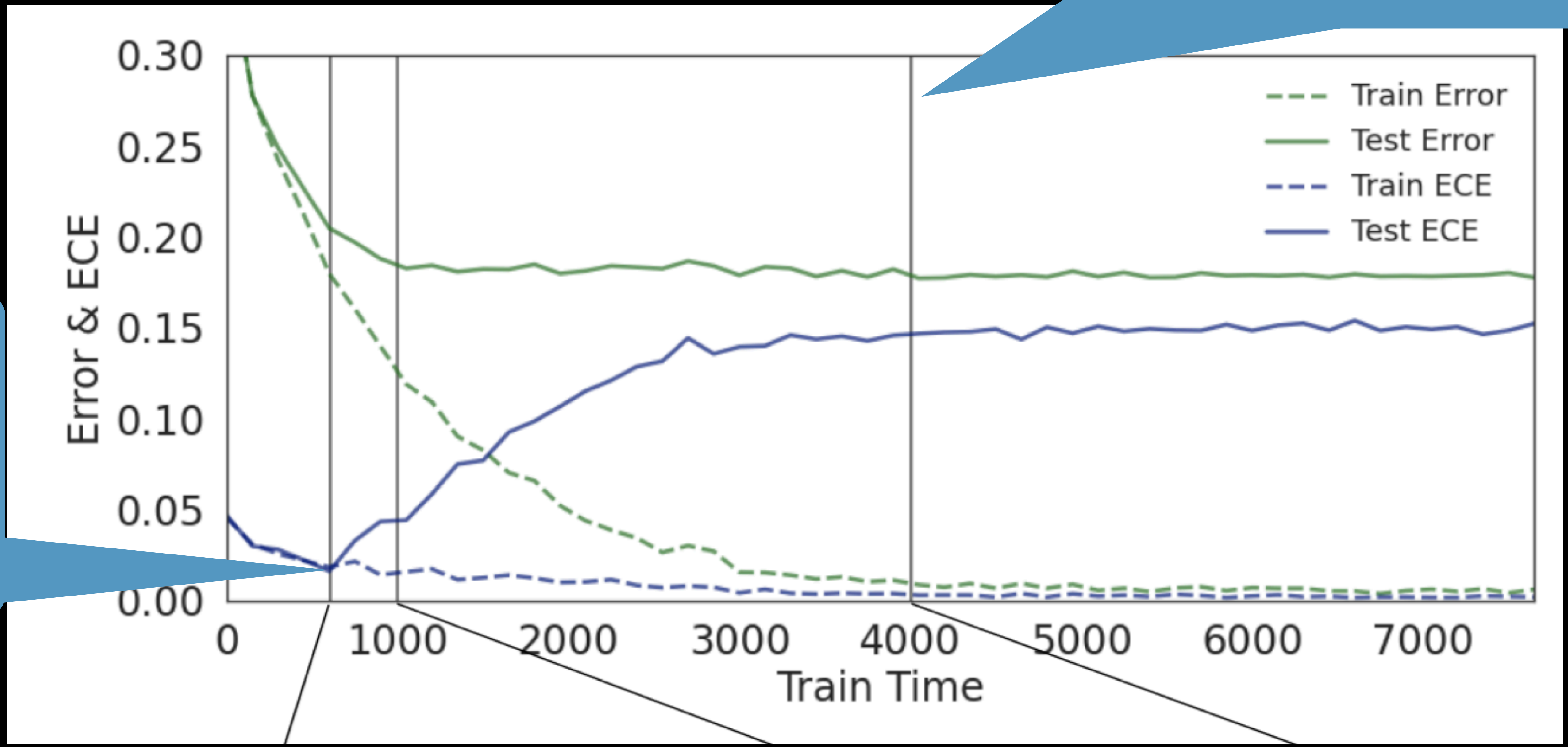
ViT on binary-CIFAR-10

End of training:
models are overconfident

Train {Error,CE} ≈ 0

Test {Error,CE} $\gg 0$

(higher=worse)



Throughout training:

Train CE ≈ 0

ViT on binary-CIFAR-10

Fundamental Decomposition

$$\underbrace{\mu\text{Test}}_{\text{Calibration Error on Test Set}} \leq \boxed{\underbrace{\mu\text{Train}}_{\text{Calibration Error on Train Set}}} + \boxed{\underbrace{|\mu\text{Test} - \mu\text{Train}|}_{\text{Calibration Generalization Gap}}}$$

** depth ≥ 2, trained with proper scoring rule, no severe augmentations, ...*

Fundamental Decomposition

$$\underbrace{\mu\text{Test}}_{\text{Calibration Error on Test Set}} \leq \boxed{\underbrace{\mu\text{Train}}_{\text{Calibration Error on Train Set}}} + \boxed{\underbrace{|\mu\text{Test} - \mu\text{Train}|}_{\text{Calibration Generalization Gap}}}$$

Empirical Claim 1

For almost all* DNNs

$$\mu\text{Train} \approx 0$$

** depth ≥ 2, trained with proper scoring rule, no severe augmentations, ...*

Fundamental Decomposition

$$\underbrace{\mu_{\text{Test}}}_{\text{Calibration Error on Test Set}} \leq \boxed{\underbrace{\cancelto{0}{\mu_{\text{Train}}}}_{\text{Calibration Error on Train Set}}} + \boxed{\underbrace{|\mu_{\text{Test}} - \mu_{\text{Train}}|}_{\text{Calibration Generalization Gap}}}$$

Empirical Claim 1

For almost all* DNNs

$$\mu_{\text{Train}} \approx 0$$

* *depth ≥ 2, trained with proper scoring rule, no severe augmentations, …*

Fundamental Decomposition

$$\underbrace{\mu_{\text{Test}}}_{\text{Calibration Error on Test Set}} \leq \boxed{\underbrace{\cancel{\mu_{\text{Train}}}}_{\text{Calibration Error on Train Set}}}^{\,0} + \boxed{\underbrace{|\mu_{\text{Test}} - \mu_{\text{Train}}|}_{\text{Calibration Generalization Gap}}}$$

Empirical Claim 1

For almost all* DNNs

$$\mu_{\text{Train}} \approx 0$$

Even when underfitting!

* depth ≥ 2, trained with proper scoring rule, no severe augmentations, …

Fundamental Decomposition

$$\underbrace{\mu_{\text{Test}}}_{\text{Calibration Error on Test Set}} \leq \underbrace{\boxed{\cancelto{0}{\mu_{\text{Train}}}}}_{\text{Calibration Error on Train Set}} + \underbrace{\boxed{\left|\mu_{\text{Test}} - \mu_{\text{Train}}\right|}}_{\text{Calibration Generalization Gap}}$$

Empirical Claim 1

For almost all* DNNs

$$\mu_{\text{Train}} \approx 0$$

Empirical Claim 2

For almost all* DNNs

$$\left|\mu_{\text{Test}} - \mu_{\text{Train}}\right| \leq \left|\text{TestError} - \text{TrainError}\right|$$

Even when underfitting!

** depth ≥ 2, trained with proper scoring rule, no severe augmentations, …*

Fundamental Decomposition

$$\underbrace{\mu_{\text{Test}}}_{\text{Calibration Error on Test Set}} \leq \underbrace{\boxed{\cancelto{0}{\mu_{\text{Train}}}}}_{\text{Calibration Error on Train Set}} + \underbrace{\boxed{\cancelto{\leq \text{Error Generalization Gap}}{\left|\mu_{\text{Test}} - \mu_{\text{Train}}\right|}}}_{\text{Calibration Generalization Gap}}$$

Empirical Claim 1

For almost all* DNNs

$$\mu_{\text{Train}} \approx 0$$

Empirical Claim 2

For almost all* DNNs

$$\left|\mu_{\text{Test}} - \mu_{\text{Train}}\right| \leq \left|\text{TestError} - \text{TrainError}\right|$$

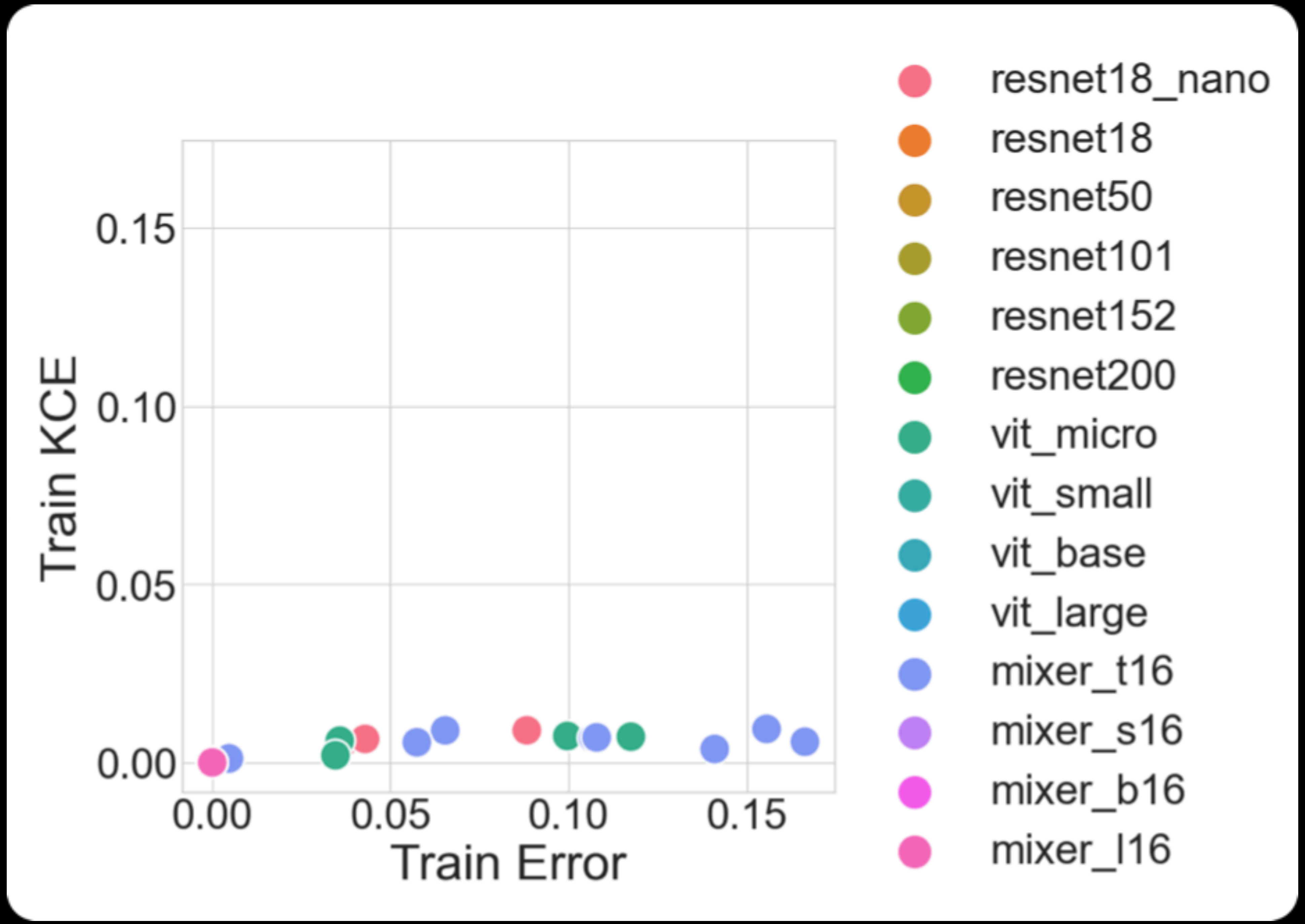
Even when underfitting!

** depth ≥ 2, trained with proper scoring rule, no severe augmentations, ...*

Empirical Claim 1

For almost all* DNNs

$$\mu_{\text{Train}} \approx 0$$

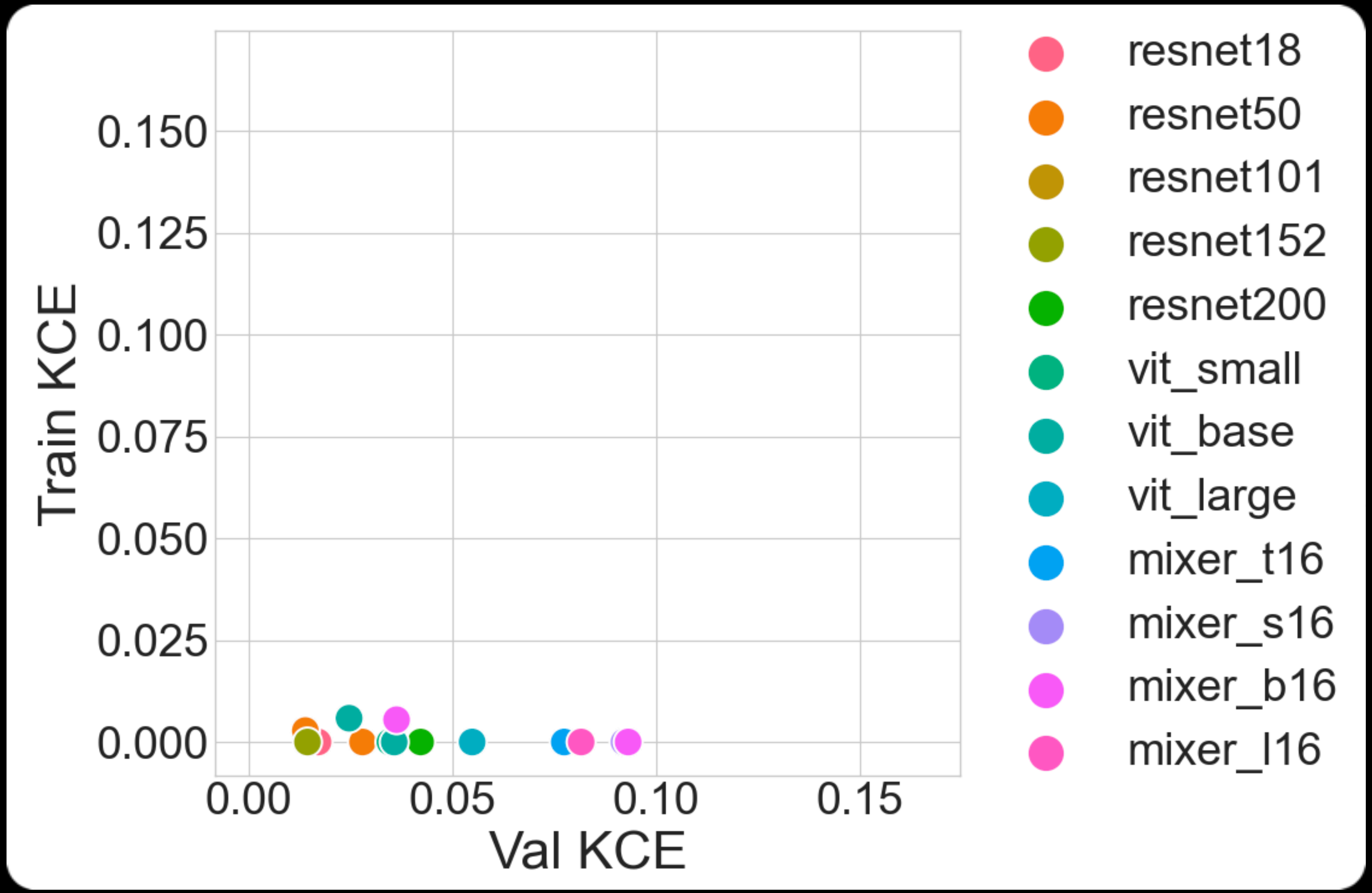


(ResNets on binary-ImageNet)

Empirical Claim 1

For almost all* DNNs

$$\mu_{\text{Train}} \approx 0$$

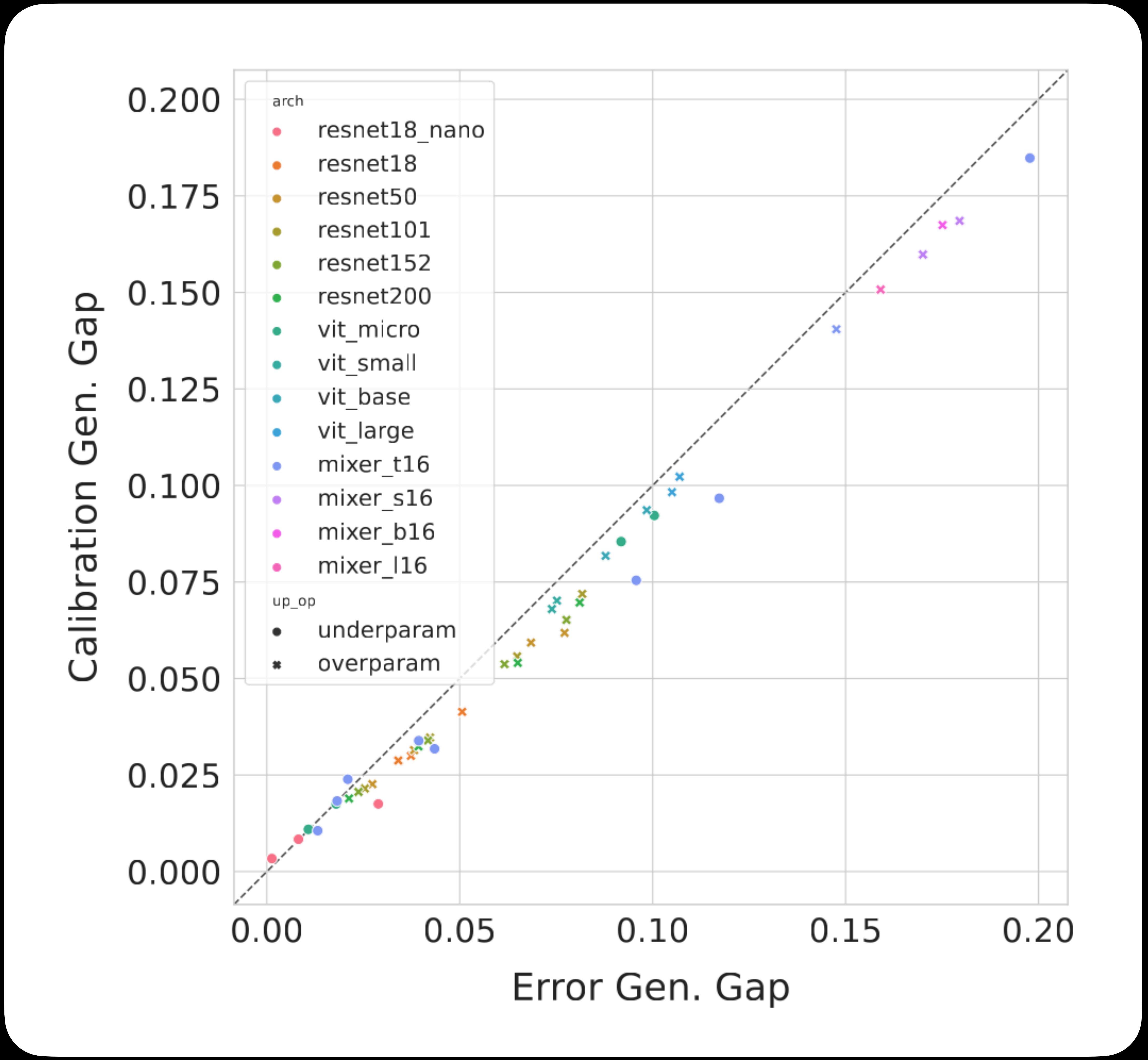


(binary-ImageNet)

Empirical Claim 2

For almost all* DNNs

$|\mu_{\text{Test}} - \mu_{\text{Train}}| \leq |\text{TestError} - \text{TrainError}|$



(binary-ImageNet)

Takeaways

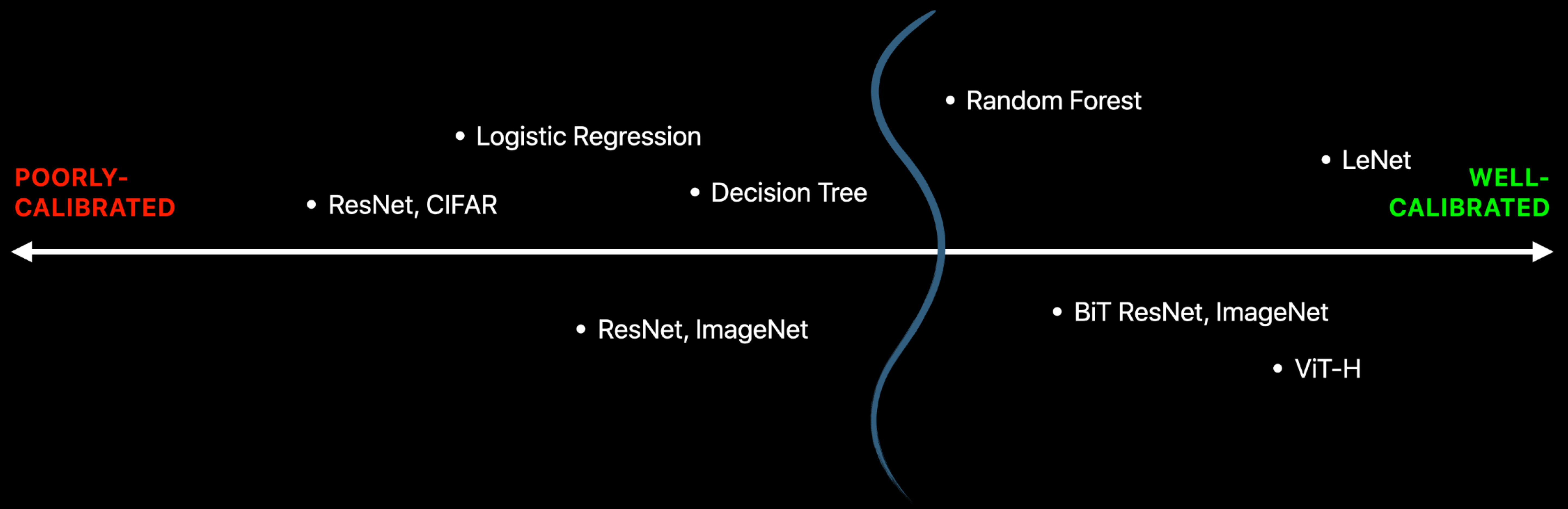
$$\text{(Test Calibration Error)} \quad \lessapprox \quad |\text{Train Error - Test Error}|$$

“Models with small generalization-gap are typically well-calibrated”

Takeaways

(Test Calibration Error) $\lessapprox$ |Train Error - Test Error|

"Models with small generalization-gap are typically well-calibrated"



Takeaways

(Test Calibration Error) $\lessapprox$ |Train Error - Test Error|

"Models with small generalization-gap are typically well-calibrated"

The following are well-calibrated:

1. Small models, on large data-sets (e.g. large vision models)
2. All models trained for 1-epoch (e.g. LLMs)

Applications

$$\underbrace{\mu_{\text{Test}}}_{\text{Calibration on Test Set}} \leq \underbrace{\mu_{\text{Train}}}_{\text{Calibration on Train Set}} + \underbrace{\left|\mu_{\text{Test}} - \mu_{\text{Train}}\right|}_{\text{Calibration Generalization Gap}}$$

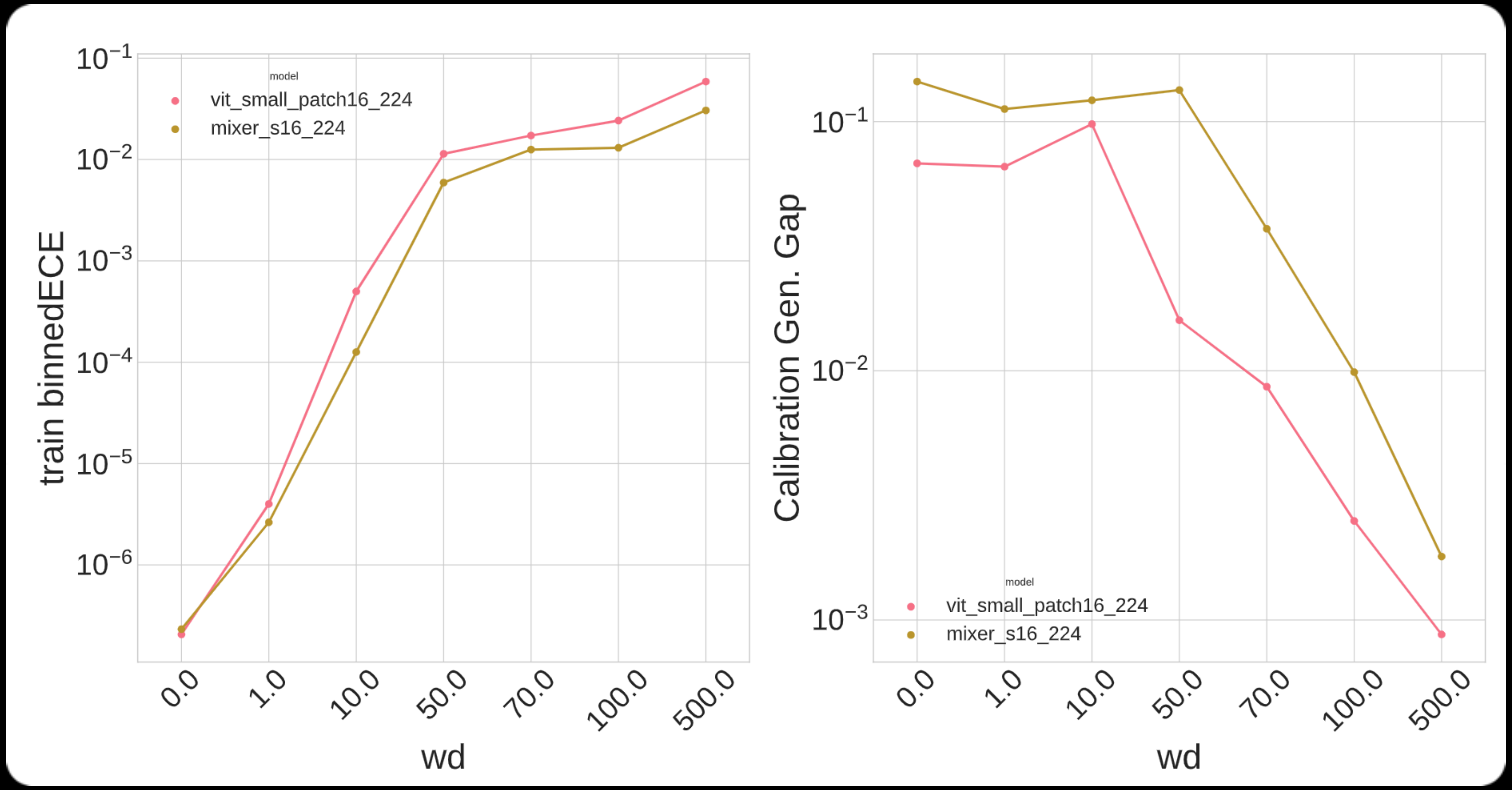
For any intervention (changing the augmentation, regularizer, etc), study its effect on:

(1) Train calibration

(2) Generalization gap

Applications: Regularization Strength

$$\underbrace{\mu_{\text{Test}}}_{\text{Calibration on Test Set}} \leq \underbrace{\mu_{\text{Train}}}_{\text{Calibration on Train Set}} + \underbrace{|\mu_{\text{Test}} - \mu_{\text{Train}}|}_{\text{Calibration Generalization Gap}}$$



Applications: Data Augmentation

$$\underbrace{\mu_{\text{Test}}}_{\text{Calibration on Test Set}} \leq \underbrace{\mu_{\text{Train}}}_{\text{Calibration on Train Set}} + \underbrace{|\mu_{\text{Test}} - \mu_{\text{Train}}|}_{\text{Calibration Generalization Gap}}$$

"Standard" data-augmentation (measure-preserving):

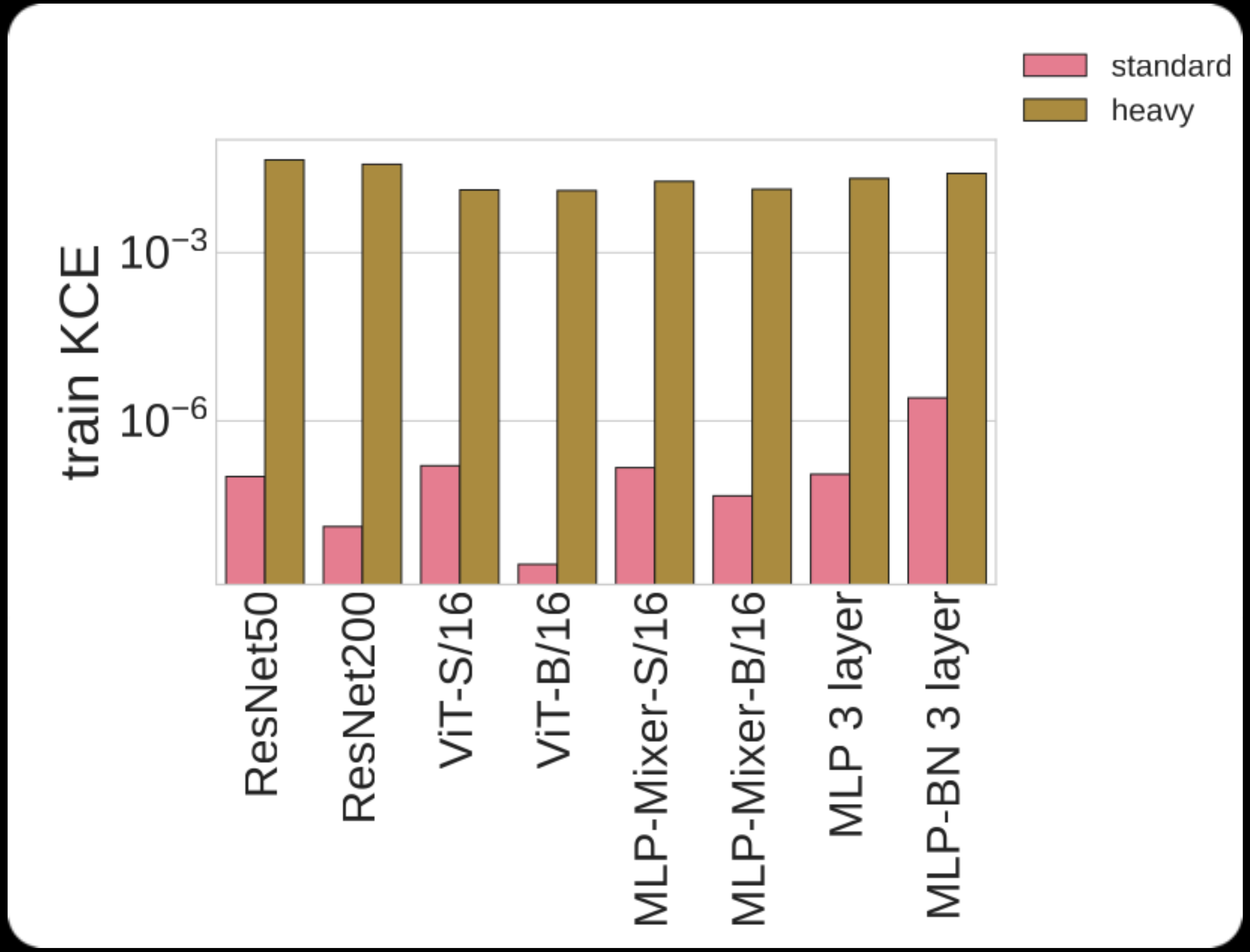
- Same TrainCE; Shrinks generalization gap

"Exotic" data-augmentation:

- *Increases TrainCE*; Shrinks generalization gap

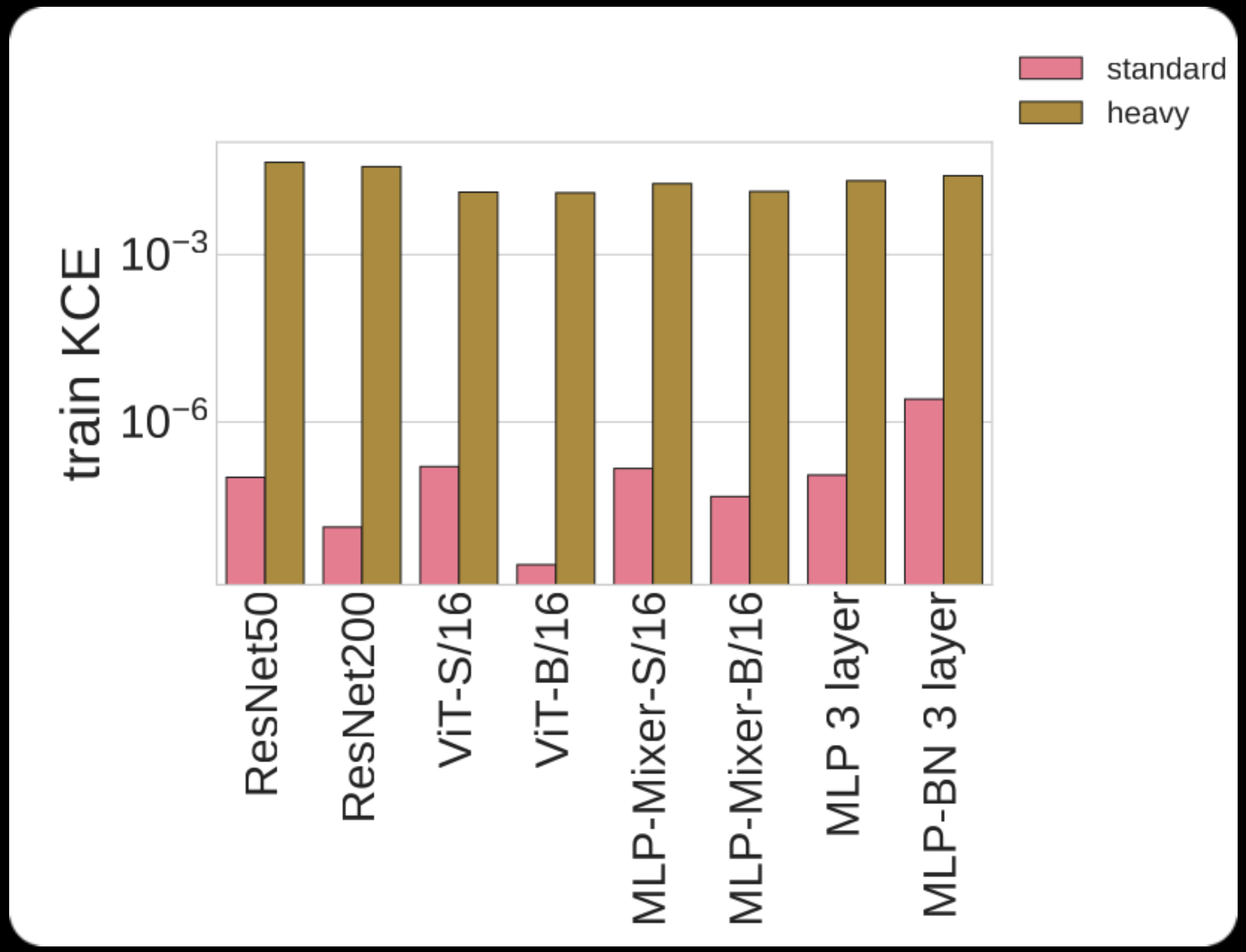
Applications: Data Augmentation

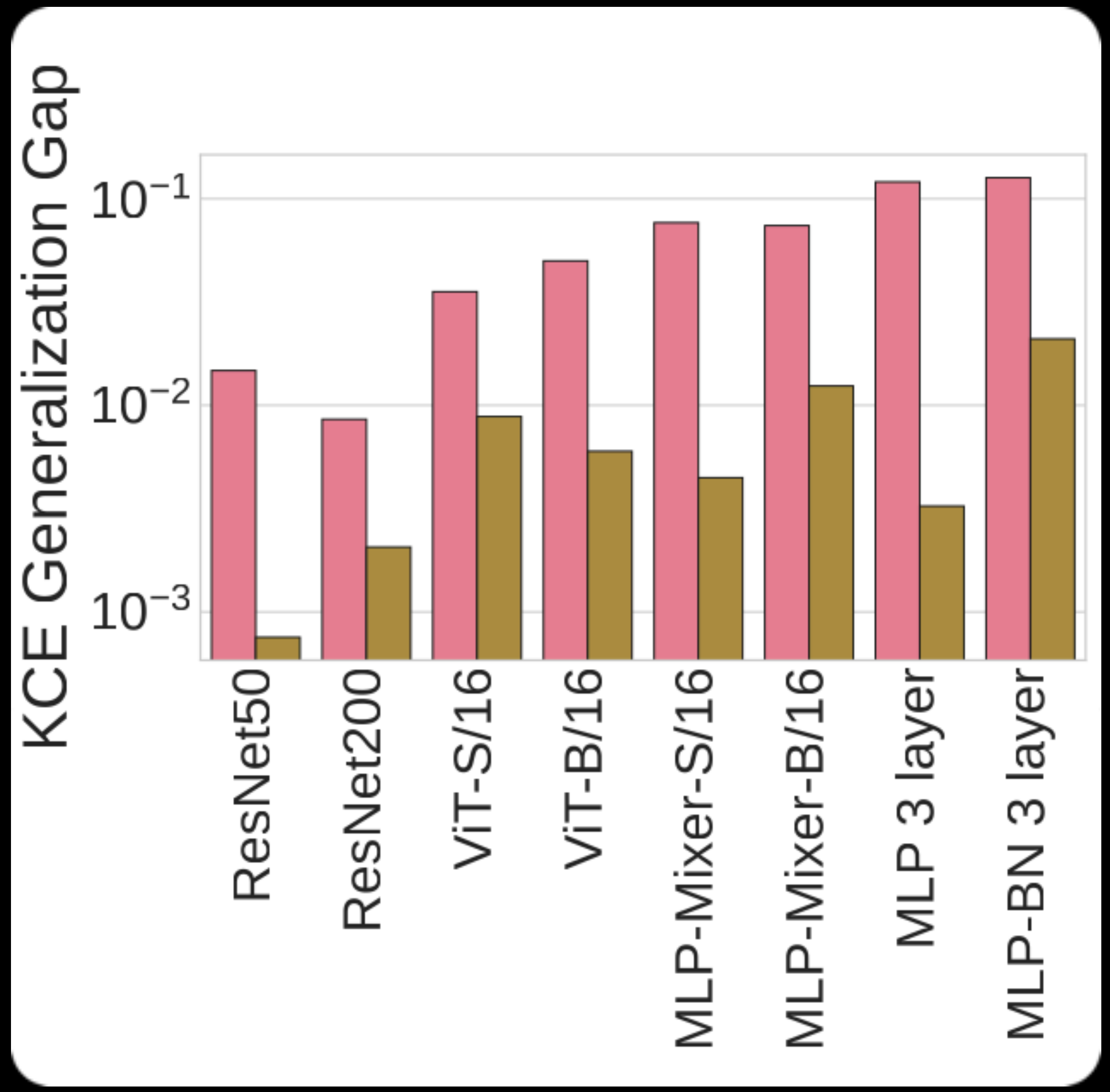
$$\underbrace{\mu_{\text{Test}}}_{\text{Calibration on Test Set}} \leq \underbrace{\mu_{\text{Train}}}_{\text{Calibration on Train Set}} + \underbrace{\left|\mu_{\text{Test}} - \mu_{\text{Train}}\right|}_{\text{Calibration Generalization Gap}}$$



Applications: Data Augmentation

$$\underbrace{\mu_{\text{Test}}}_{\text{Calibration on Test Set}} \leq \underbrace{\mu_{\text{Train}}}_{\text{Calibration on Train Set}} + \underbrace{|\mu_{\text{Test}} - \mu_{\text{Train}}|}_{\text{Calibration Generalization Gap}}$$





Part 3.
Theory

When are Claims 1 & 2 provably true?

Empirical Claim 2

For almost all* DNNs

$$|\mu_{\text{Test}} - \mu_{\text{Train}}| \leq |\text{TestError} - \text{TrainError}|$$

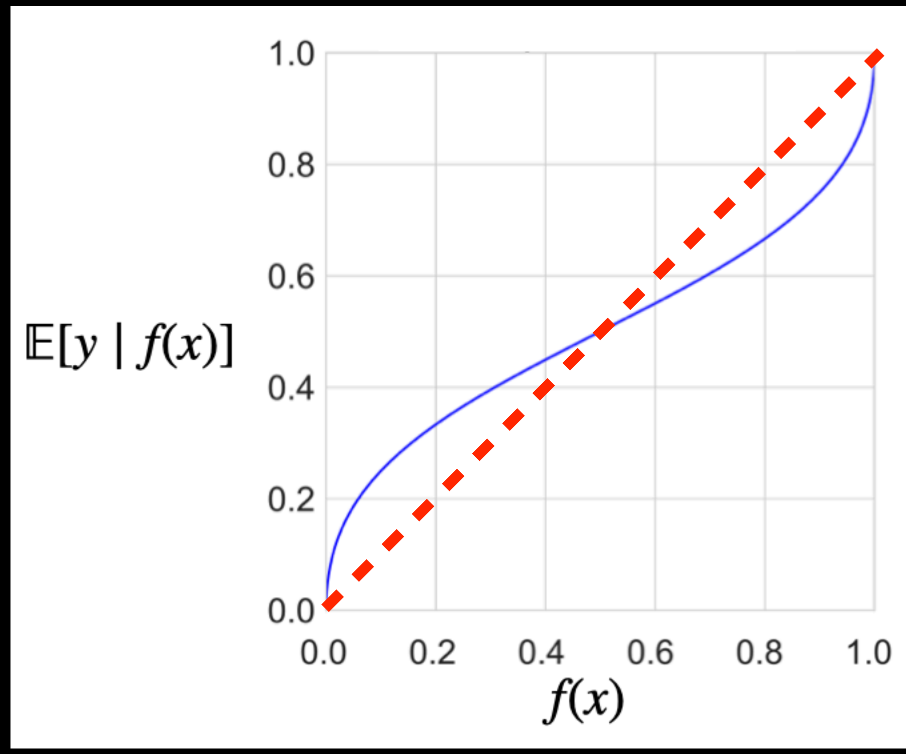
Empirical Claim 2

For almost all* DNNs

$|\mu_{\text{Test}} - \mu_{\text{Train}}| \leq |\text{TestError} - \text{TrainError}|$

Assumption 1: f is overconfident on test.

$$\forall \ell \in [0,1] : \underset{\mathbb{D}_{\text{test}}}{\mathbb{E}} [\text{Acc}(f,y) \mid f = \ell] \leq \text{Conf}(v)$$



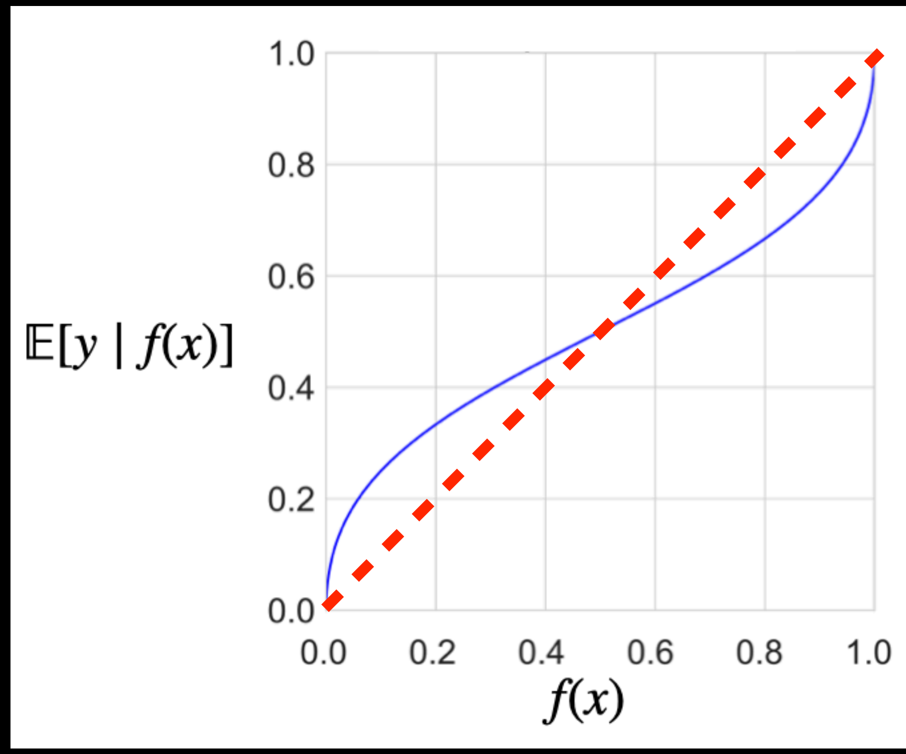
Empirical Claim 2

For almost all* DNNs

$|\mu_{\text{Test}} - \mu_{\text{Train}}| \leq |\text{TestError} - \text{TrainError}|$

Assumption 1: f is overconfident on test.

$$\forall \ell \in [0,1] : \mathop{\mathbb{E}}_{\mathbb{D}_{\text{test}}} [\text{Acc}(f, y) \mid f = \ell] \leq \text{Conf}(v)$$



Assumption 2: f is more confident on train than on test.

$$\mathop{\mathbb{E}}_{\mathbb{D}_{\text{train}}} [\text{Conf}(f)] \geq \mathop{\mathbb{E}}_{\mathbb{D}_{\text{test}}} [\text{Conf}(f)]$$

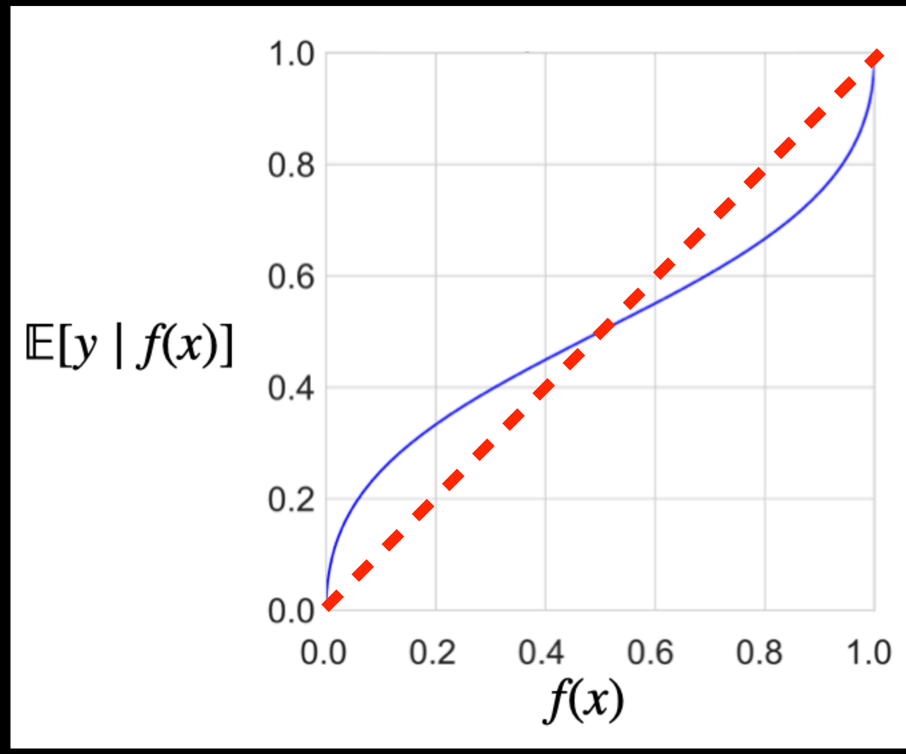
Empirical Claim 2

For almost all* DNNs

$|\mu_{\text{Test}} - \mu_{\text{Train}}| \leq |\text{TestError} - \text{TrainError}|$

Assumption 1: f is overconfident on test.

$$\forall \ell \in [0,1] : \mathop{\mathbb{E}}_{\mathbb{D}_{\text{test}}} [\text{Acc}(f, y) \mid f = \ell] \leq \text{Conf}(v)$$



Assumption 2: f is more confident on train than on test.

$$\mathop{\mathbb{E}}_{\mathbb{D}_{\text{train}}} [\text{Conf}(f)] \geq \mathop{\mathbb{E}}_{\mathbb{D}_{\text{test}}} [\text{Conf}(f)]$$

Theorem 2 (Calibration Generalization Bound). *Under Assumptions 2 and 3, we have*

$$\text{ECE}(\mathbb{D}_{\text{test}}) - \text{ECE}(\mathbb{D}_{\text{train}}) \leq \text{Error}(\mathbb{D}_{\text{test}}) - \text{Error}(\mathbb{D}_{\text{train}})$$

Empirical Claim 1

For almost all* DNNs

$$\mu_{\text{Train}} \approx 0$$

Given: Distribution $D = \widehat{D} = \{(x_i, y_i)\}_{i \in [n]}$

Given: Distribution $D = \widehat{D} = \{(x_i, y_i)\}_{i \in [n]}$

What we'd like to do:

Exactly minimize expected loss, over *all functions*:

$$f^* = \underset{f:\mathcal{X}\to[0,1]}{\operatorname{argmin}} \underset{x,y\sim\mathcal{D}}{\mathbb{E}}[\ell(f(x), y)]$$

Given: Distribution $D = \widehat{D} = \{(x_i, y_i)\}_{i \in [n]}$

What we'd like to do:

Exactly minimize expected loss, over
all functions:

$$f^* = \underset{f:\mathcal{X}\to[0,1]}{\operatorname{argmin}} \underset{x,y\sim\mathcal{D}}{\mathbb{E}}[\ell(f(x), y)]$$

$$\implies f^*(x) = p_{\mathcal{D}}(y|x)$$

perfectly calibrated

Given: Distribution $D = \widehat{D} = \{(x_i, y_i)\}_{i\in[n]}$

What we'd like to do:

Exactly minimize expected loss, over *all functions*:

$$f^* = \operatorname*{argmin}_{f:\mathcal{X}\to[0,1]} \mathop{\mathbb{E}}_{x,y\sim\mathcal{D}}[\ell(f(x), y)]$$

$$\implies f^*(x) = p_{\mathcal{D}}(y|x)$$

perfectly calibrated

What we actually do:

Run SGD* to *approximately minimize* expected loss, over *restricted family* $\{f_\theta : \theta \in \Theta\}$:

$$\tilde{f} = \operatorname*{SGDmin}_{\theta\in\Theta} \mathop{\mathbb{E}}_{x,y\sim\mathcal{D}}[\ell(f_\theta(x), y)]$$

Given: Distribution $D = \widehat{D} = \{(x_i, y_i)\}_{i \in [n]}$

What we'd like to do:

Exactly minimize expected loss, over *all functions*:

$$f^* = \underset{f:\mathcal{X}\to[0,1]}{\operatorname{argmin}} \underset{x,y\sim\mathcal{D}}{\mathbb{E}}[\ell(f(x), y)]$$

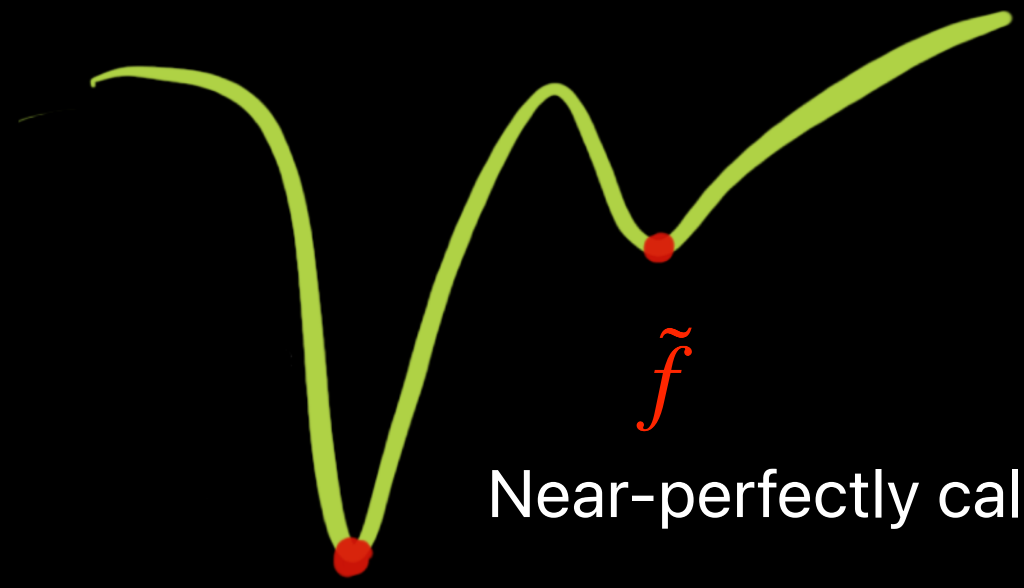
$$\implies f^*(x) = p_{\mathcal{D}}(y|x)$$

perfectly calibrated

What we actually do:

Run SGD* to *approximately minimize* expected loss, over *restricted family* $\{f_\theta : \theta \in \Theta\}$:

$$\tilde{f} = \underset{\theta\in\Theta}{\operatorname{SGDmin}} \underset{x,y\sim\mathcal{D}}{\mathbb{E}}[\ell(f_\theta(x), y)]$$



$\tilde{f}$

Near-perfectly calibrated

f^*

Perfectly calibrated
(Bayes-optimal)

Given: Distribution $D = \widehat{D} = \{(x_i, y_i)\}_{i \in [n]}$

What we'd like to do:

Exactly minimize expected loss, over *all functions*:

$$f^* = \underset{f:\mathcal{X}\to[0,1]}{\operatorname{argmin}} \underset{x,y\sim\mathcal{D}}{\mathbb{E}}[\ell(f(x), y)]$$

$$\implies f^*(x) = p_{\mathcal{D}}(y|x)$$

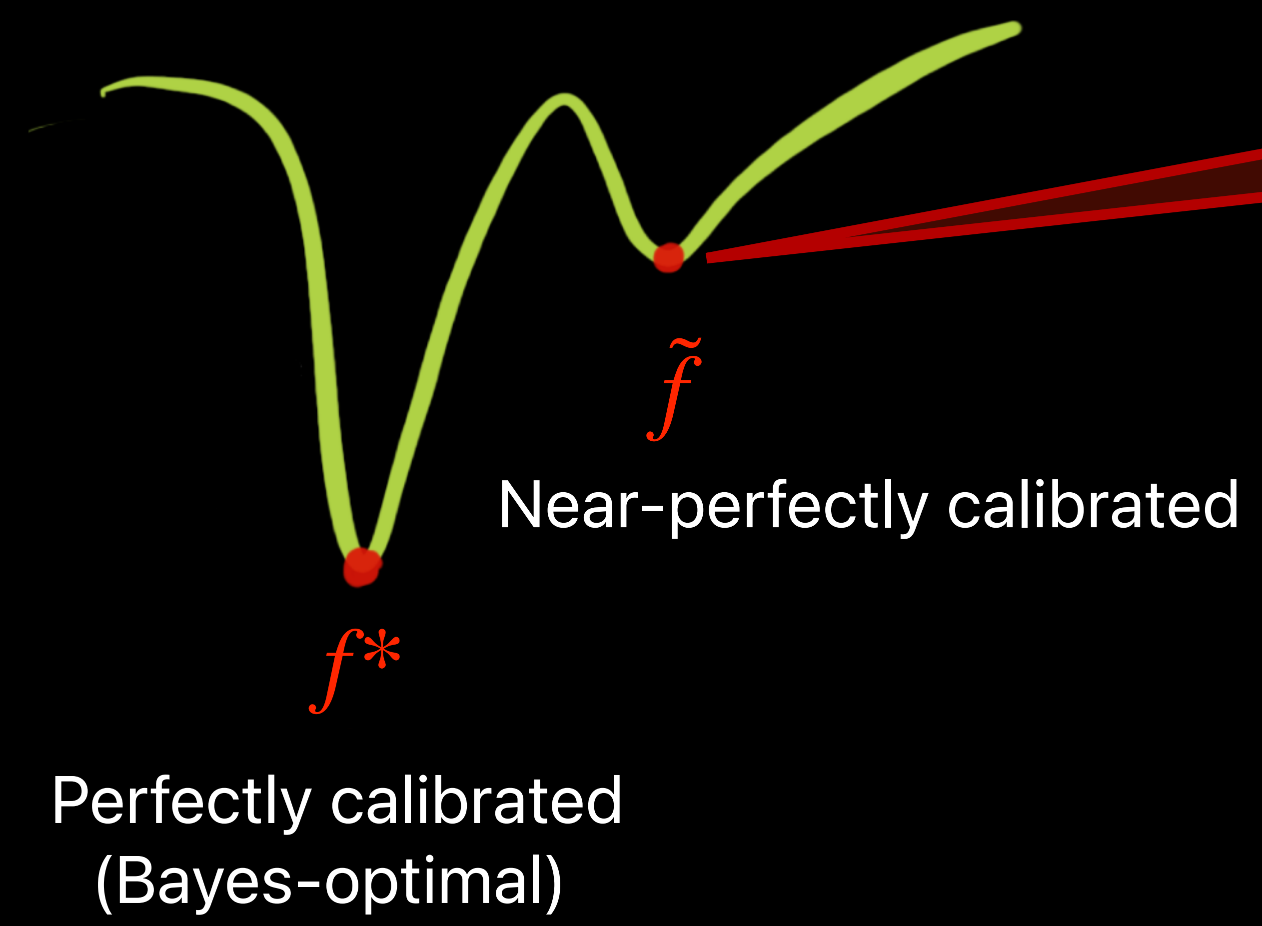
perfectly calibrated

What we actually do:

Run SGD* to *approximately minimize* expected loss, over *restricted family* $\{f_\theta : \theta \in \Theta\}$:

$$\tilde{f} = \underset{\theta\in\Theta}{\mathrm{SGDmin}} \underset{x,y\sim\mathcal{D}}{\mathbb{E}}[\ell(f_\theta(x), y)]$$

what's special about this point?



In general, when does:

suboptimal loss-minimization $\implies$ *near-optimal* calibration?

Toy Theorem

For all f, D, and proper loss ℓ, TFAE:

1. f is perfectly calibrated w.r.t. D

2. The loss of $f : \mathcal{X} \to [0,1]$ on D cannot be improved by post-processing $\kappa : [0,1] \to [0,1]$

$$\forall \kappa : [0,1] \to [0,1], \quad \mathcal{L}_D(f) \leq \mathcal{L}_D(\kappa \circ f)$$

where $\mathcal{L}_D(f) := \mathbb{E}_{x,y \sim D}[\ell(f(x), y)]$ is the expected loss

Toy Theorem

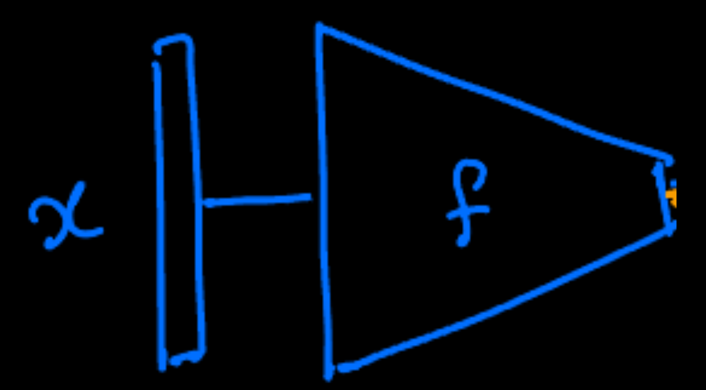
For all f, D, and proper loss ℓ, TFAE:

1. f is perfectly calibrated w.r.t. D

2. The loss of $f : \mathscr{X} \to [0,1]$ on D cannot be improved by post-processing $\kappa : [0,1] \to [0,1]$

$$\forall \kappa : [0,1] \to [0,1], \quad \mathscr{L}_D(f) \leq \mathscr{L}_D(\kappa \circ f)$$

where $\mathscr{L}_D(f) := \mathbb{E}_{x,y\sim D}[\ell(f(x), y)]$ is the expected loss



Toy Theorem

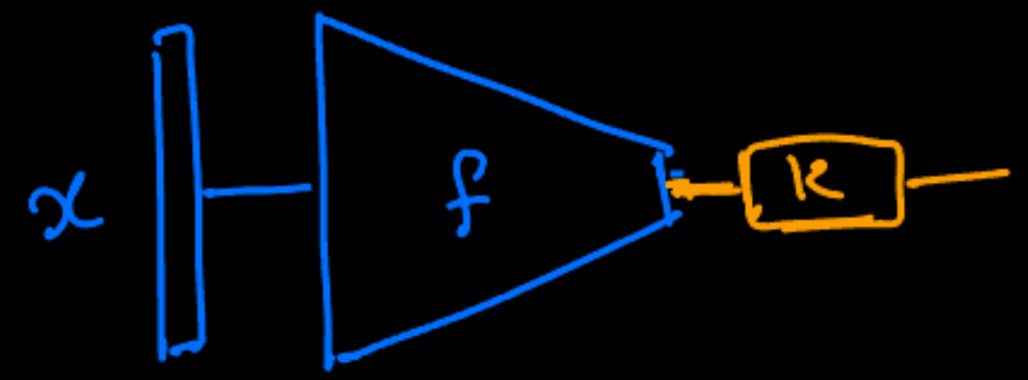
For all f, D, and proper loss ℓ, TFAE:

1. f is perfectly calibrated w.r.t. D

2. The loss of $f : \mathscr{X} \to [0,1]$ on D cannot be improved by post-processing $\kappa : [0,1] \to [0,1]$

$$\forall \kappa : [0,1] \to [0,1], \quad \mathscr{L}_D(f) \leq \mathscr{L}_D(\kappa \circ f)$$

where $\mathscr{L}_D(f) := \mathbb{E}_{x,y \sim D}[\ell(f(x), y)]$ is the expected loss



Toy Theorem

For all f, D, and proper loss ℓ, TFAE:

1. f is perfectly calibrated w.r.t. D

2. The loss of f on D cannot be improved by post-processing:

$$\forall \kappa : \mathbb{R} \to \mathbb{R}, \quad \mathscr{L}_D(f) \leq \mathscr{L}_D(\kappa \circ f)$$

Suggestive properties:

1. Requires only "weak local-optimality", not global optimality

2. Post-processing can be represented by adding a layer

Toy Theorem

For all f, D, and proper loss ℓ, TFAE:

1. f is perfectly calibrated w.r.t. D

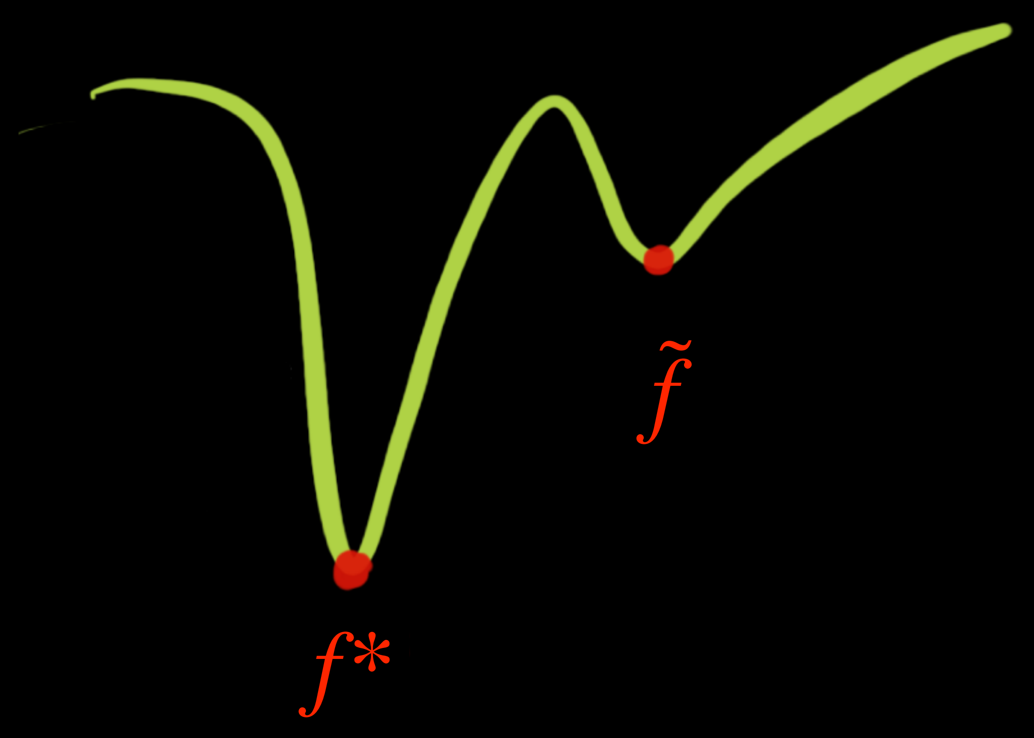
2. The loss of f on D cannot be improved by post-processing:

$$\forall \kappa : \mathbb{R} \to \mathbb{R}, \quad \mathscr{L}_D(f) \leq \mathscr{L}_D(\kappa \circ f)$$

Suggestive properties:

1. Requires only "weak local-optimality", not global optimality

2. Post-processing can be represented by adding a layer



Toy Theorem

For all f, D, and proper loss ℓ, TFAE:

1. f is perfectly calibrated w.r.t. D

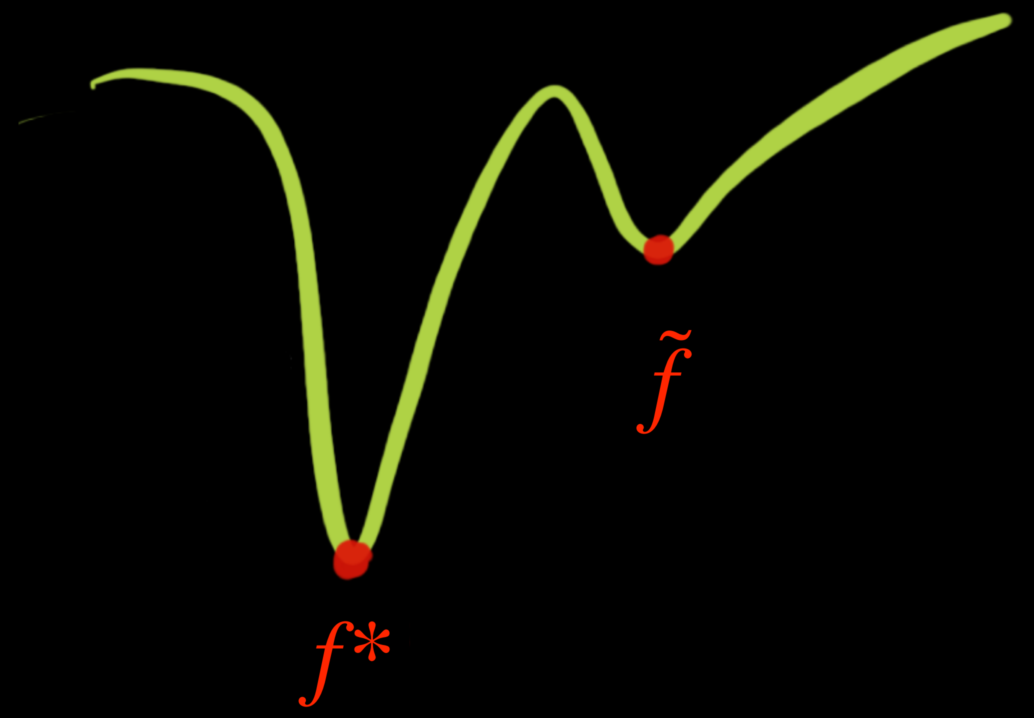
2. The loss of f on D cannot be improved by post-processing:

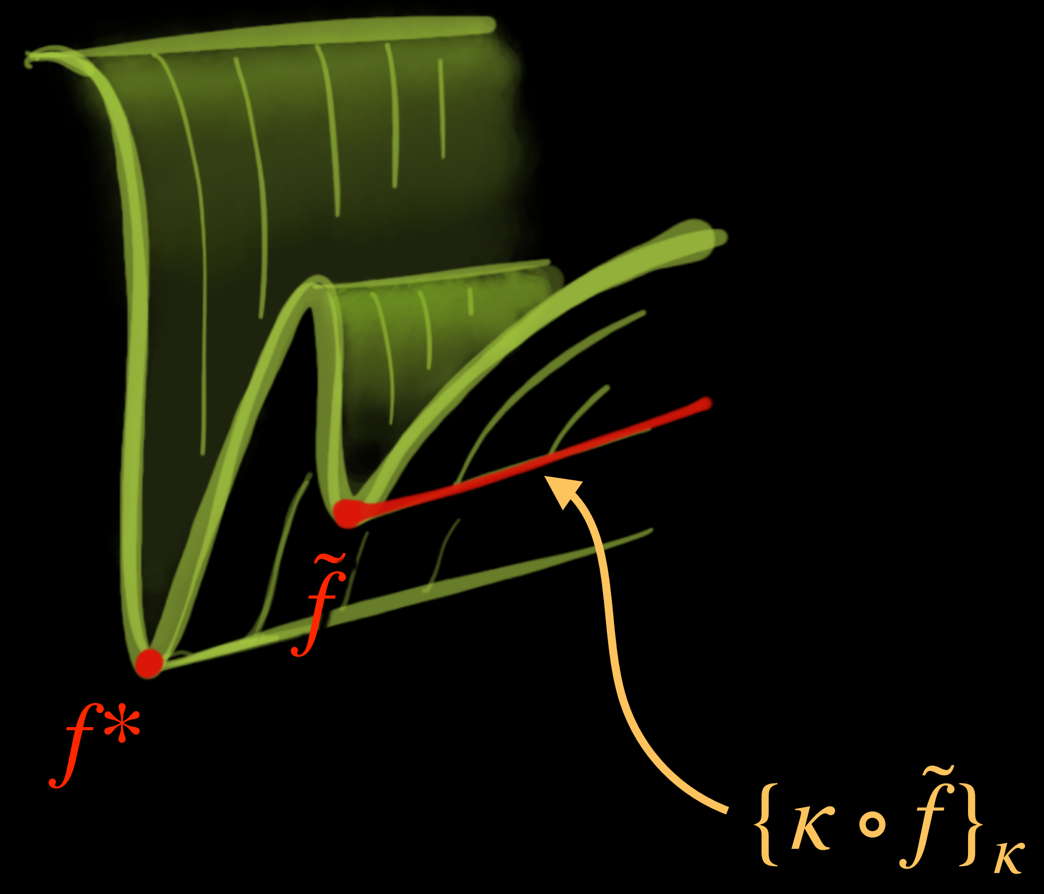
$$\forall \kappa : \mathbb{R} \to \mathbb{R}, \quad \mathscr{L}_D(f) \leq \mathscr{L}_D(\kappa \circ f)$$

Suggestive properties:

1. Requires only "weak local-optimality", not global optimality

2. Post-processing can be represented by adding a layer





Toy Theorem

For all f, D, and proper loss ℓ, TFAE:

1. f is perfectly calibrated w.r.t. D

2. The loss of f on D cannot be improved by post-processing:

$$\boxed{\forall \kappa : \mathbb{R} \to \mathbb{R},} \ \mathscr{L}_D(f) \leq \mathscr{L}_D(\kappa \circ f)$$

Problems:

1. Only characterizes perfect calibration

2. Requires composition with arbitrary functions
   (not just "nice" ones that can be represented by NNs)

Toy Theorem

"f is perfectly calibrated iff its loss can't be improved at all by post-processing with an arbitrary function"

Toy Theorem

"f is perfectly calibrated iff its loss can't be improved at all by post-processing with an arbitrary function"

Dream Theorem

"f is close to calibrated iff its loss can't be improved much by post-processing with a smooth function"

Toy Theorem	*"f is perfectly calibrated iff its loss can't be improved at all by post-processing with an arbitrary function"*
Dream Theorem	*"f is close to calibrated iff its loss can't be improved much by post-processing with a smooth function"*

How to formalize "close to"? Calibration distance $\mathrm{dCE}(f)$!

Toy Theorem | "f is perfectly [test] calibrated iff its [test] loss can't be improved at all by post-processing with an arbitrary function"

Dream Theorem | "f is close to calibrated iff its loss can't be improved much by post-processing with a smooth function"

How to formalize "close to"? Calibration distance $\mathrm{dCE}(f)$!

Dream Theorem

"f is close to calibrated iff its loss can't be improved much by post-processing with a smooth function"

Dream Theorem

"f is close to calibrated iff its loss can't be improved much by post-processing with a smooth function"

Theorem

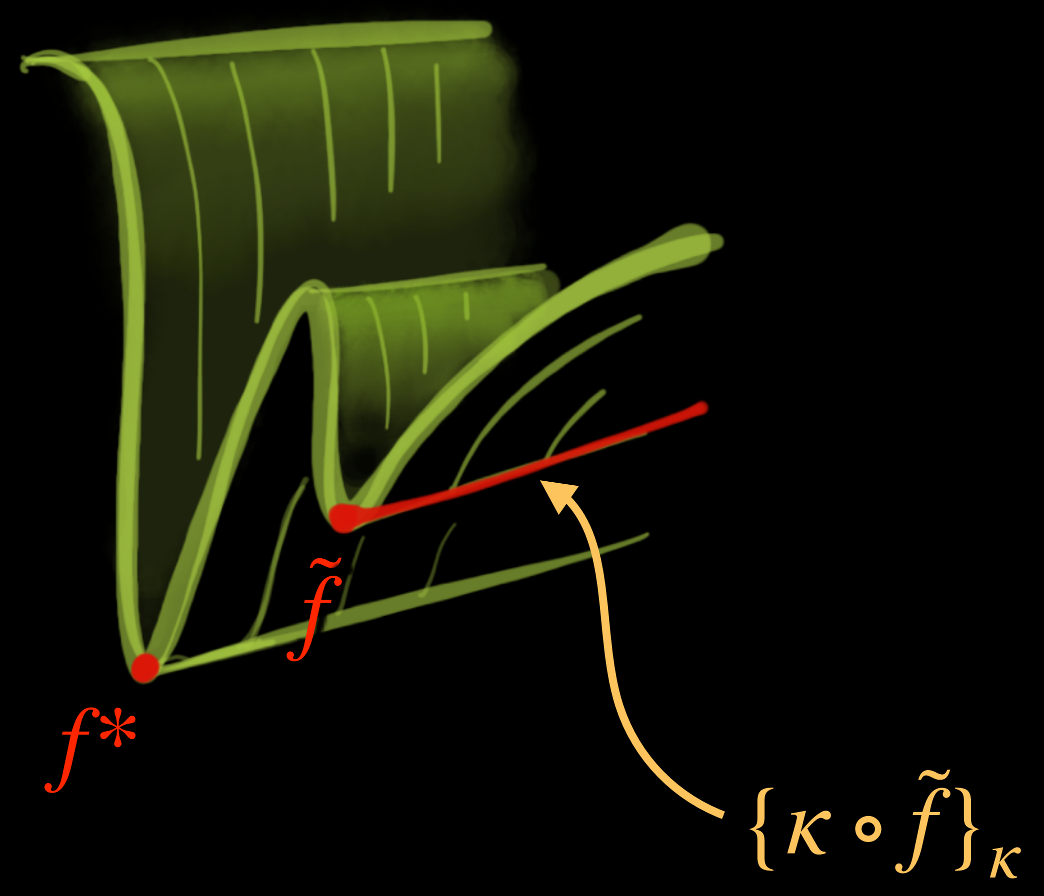
(distance from calibration) ~ poly*(potential post-processing improvement)*

Dream Theorem

"f is close to calibrated iff its loss can't be improved much by post-processing with a smooth function"

Theorem

(distance from calibration) ~ poly*(potential post-processing improvement)*



Dream Theorem

"f is close to calibrated iff its loss can't be improved much by post-processing with a smooth function"

Theorem

(distance from calibration) ~ poly(*potential post-processing improvement*)

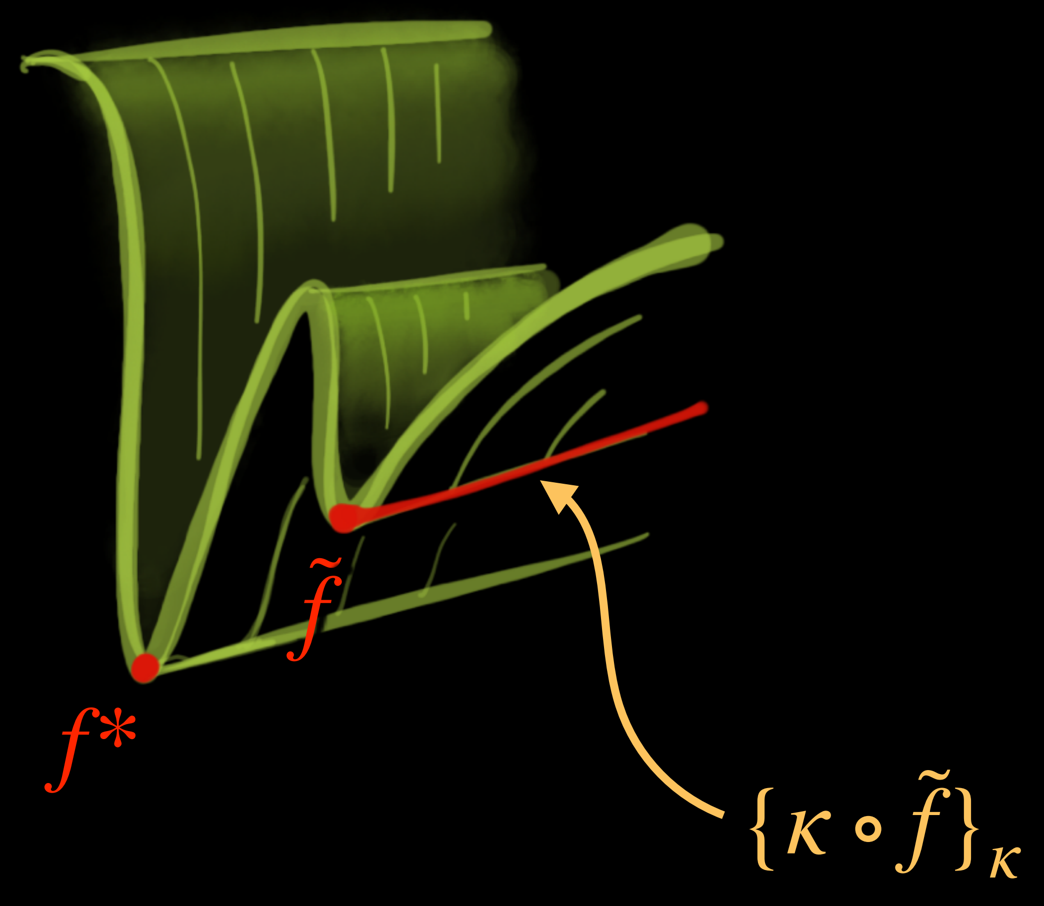
Theorem 1.3. *There exist constants $c_1, c_2 > 0$ such that for all predictors $f : \mathcal{X} \to [0,1]$ and all distributions $\mathcal{D}$, the following holds.*

Let K denote the family of all post-processing functions $\kappa : [0,1] \to [0,1]$ such that the update function $\eta(f) = \kappa(f) - f$ is 1-Lipschitz. Define the "gap calibration error" of f as the maximum improvement in MSE loss via post-processings in K:

$$\mathrm{gapCE}(f) = \mathrm{MSE}_{\mathcal{D}}(f) - \min_{\kappa \in K} \mathrm{MSE}_{\mathcal{D}}(\kappa \circ f).$$

Then, the maximum loss improvement (gapCE) polynomially bounds the distance from calibration (dCE):

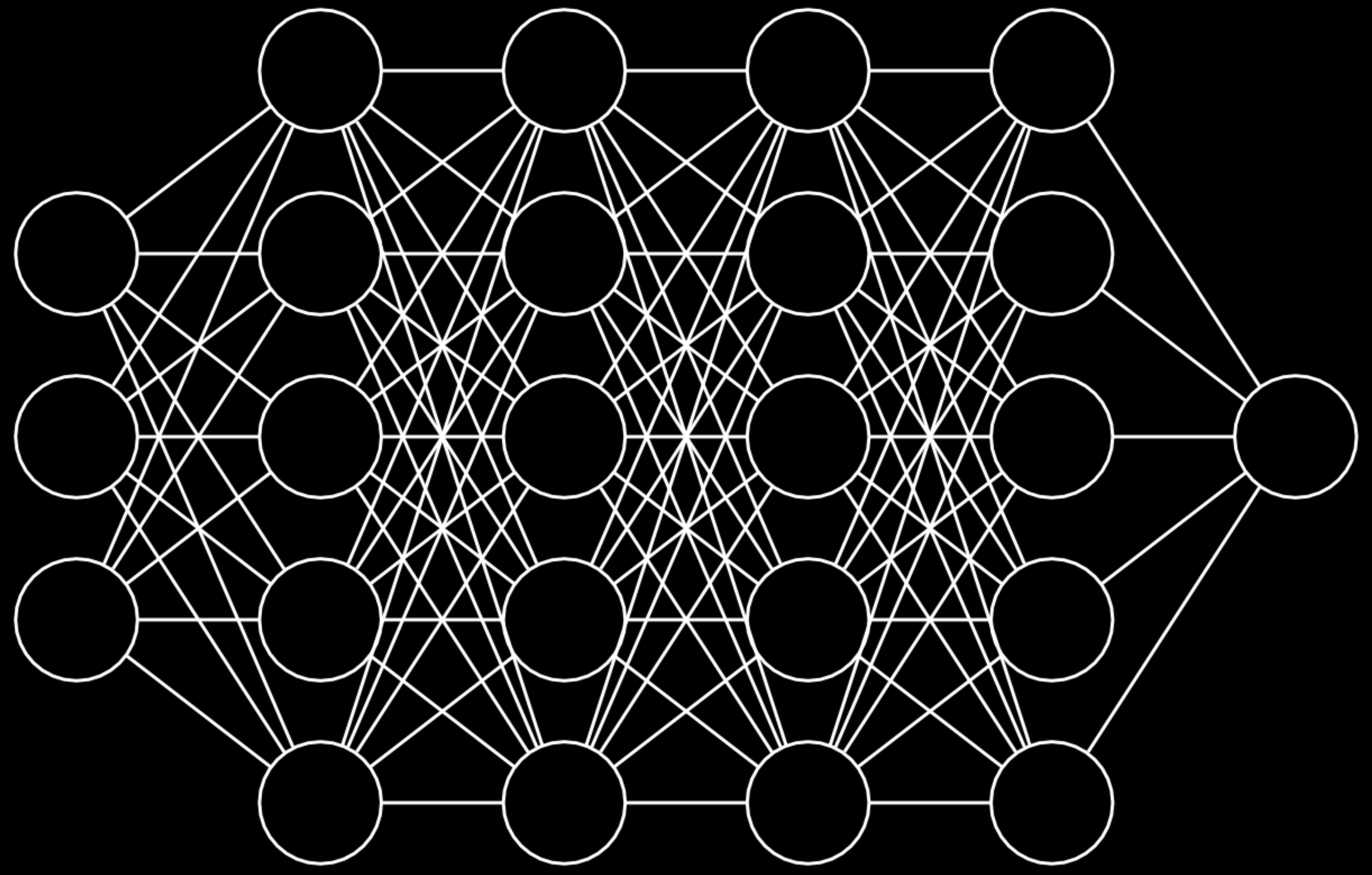
$$c_1\, \mathsf{dCE}(f)^4 \le \mathrm{gapCE}(f) \le c_2\, \mathsf{dCE}(f).$$

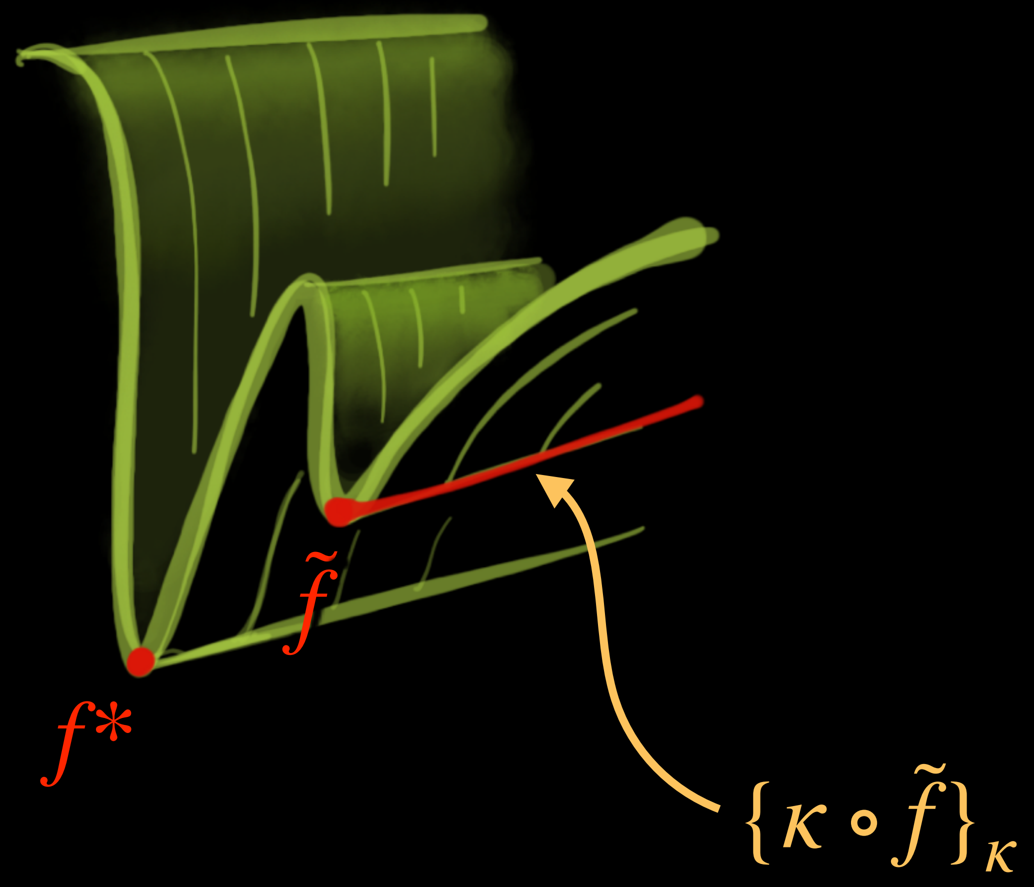


When is the “no loss improvement” condition satisfied?

1. **(Algorithmic assumption):**
   If it were possible to improve loss via a simple post-processing, SGD would have done it already.

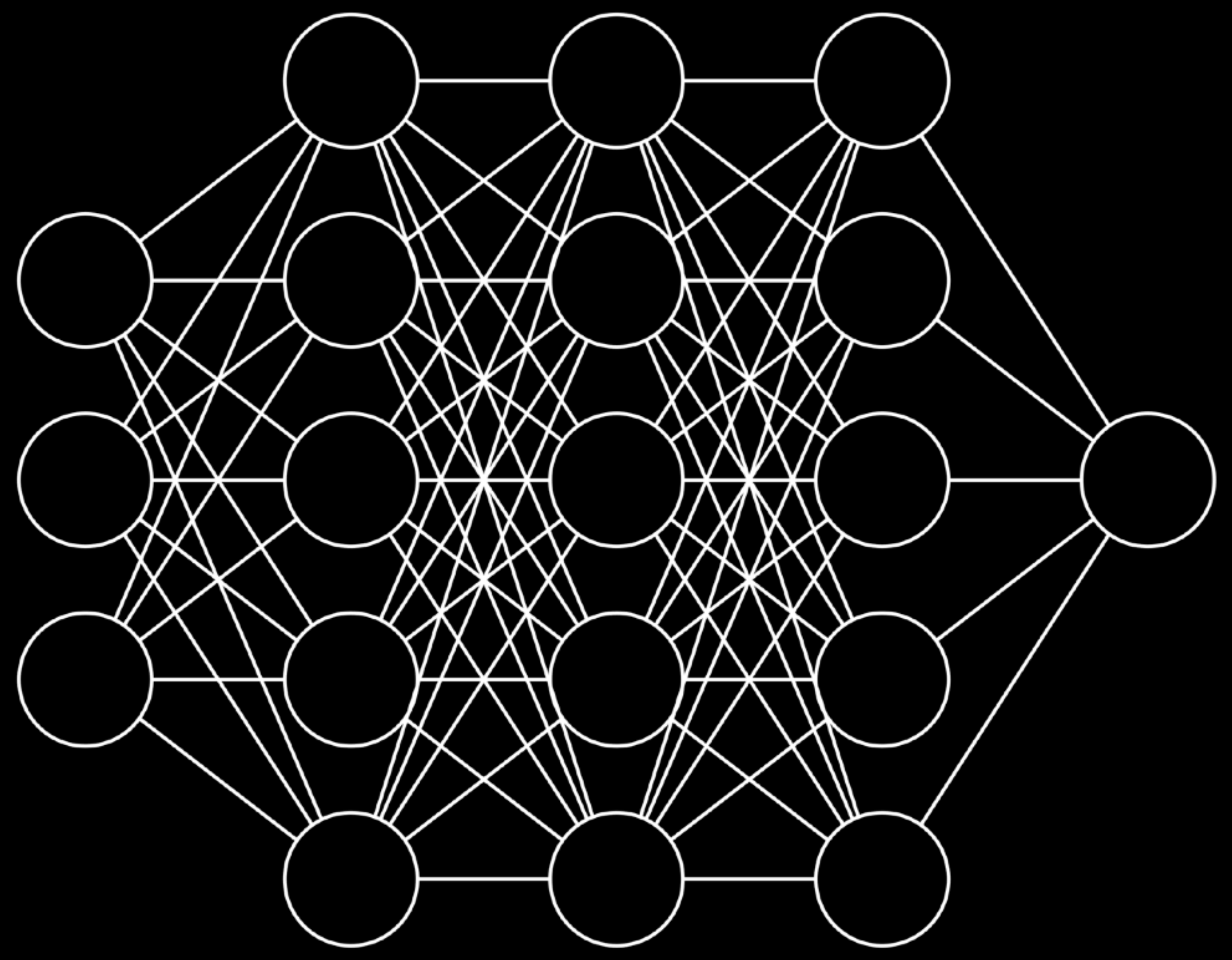
$f \mapsto \kappa \circ f$ *is a “simple” update for SGD on deep nets*





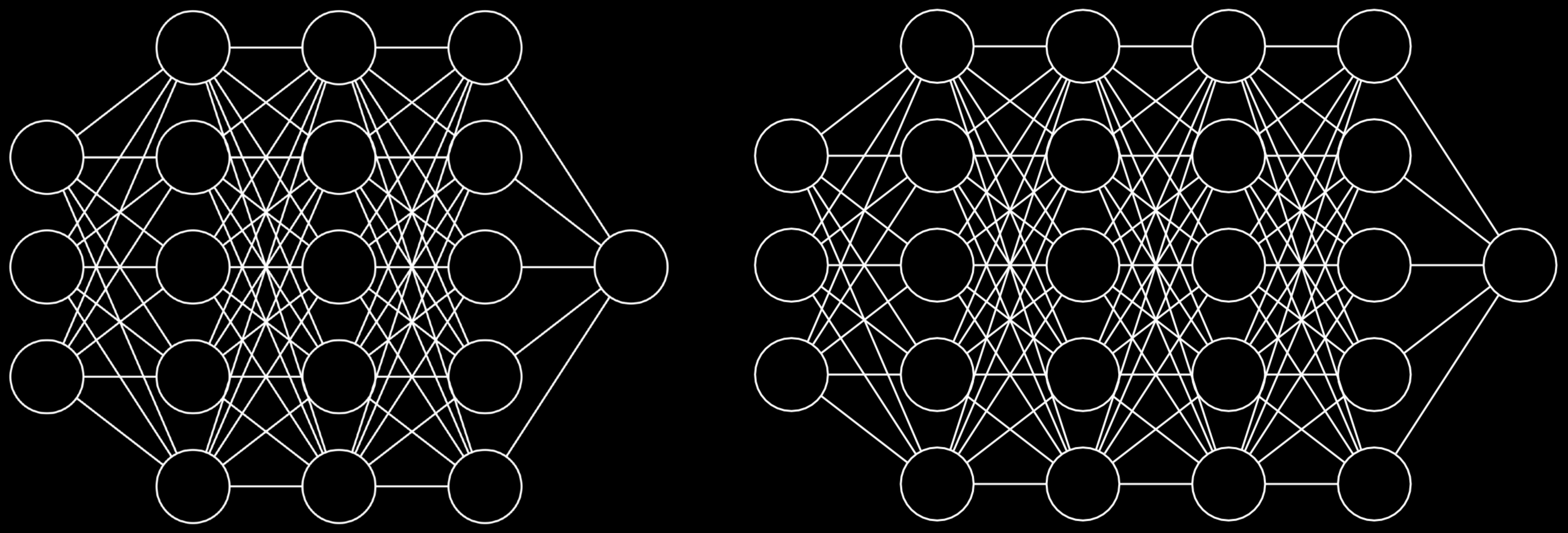
When is the “no loss improvement” condition satisfied?

2. **(Human-in-loop assumption):**
   If it were possible to add a layer, train it optimally, and improve the loss, then the human trainer would have done it already



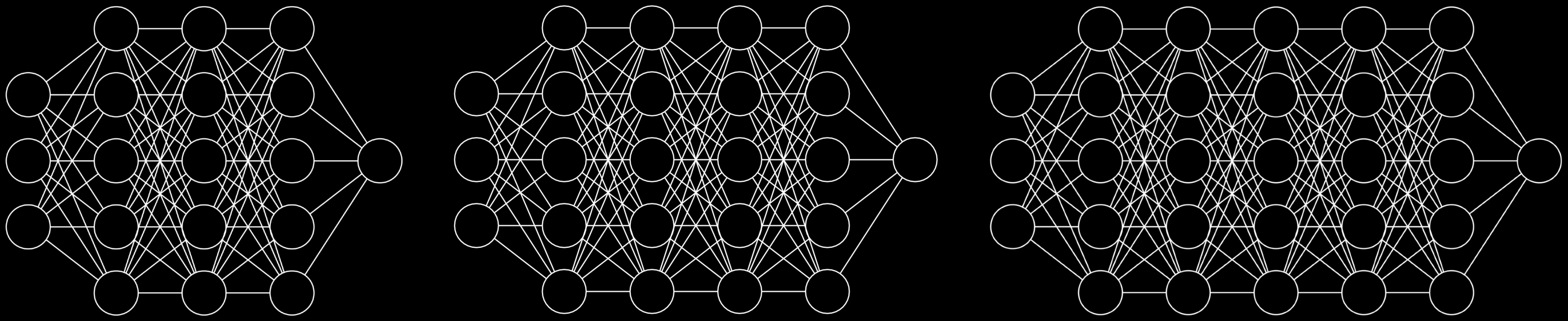
When is the “no loss improvement” condition satisfied?

2. **(Human-in-loop assumption):**
   If it were possible to add a layer, train it optimally, and improve the loss, then the human trainer would have done it already



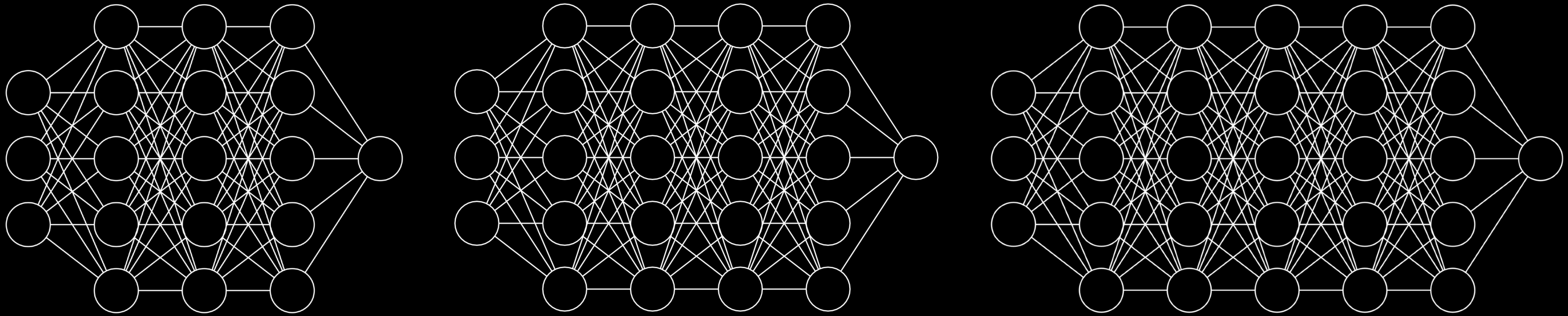
When is the “no loss improvement” condition satisfied?

2. **(Human-in-loop assumption):**
   If it were possible to add a layer, train it optimally, and improve the loss, then the human trainer would have done it already



When is the “no loss improvement” condition satisfied?

2. **(Human-in-loop assumption):**
   If it were possible to add a layer, train it optimally, and improve the loss, then the human trainer would have done it already $\Longrightarrow$ output of human is “nearly post-processing optimal”



When is the “no loss improvement” condition satisfied?

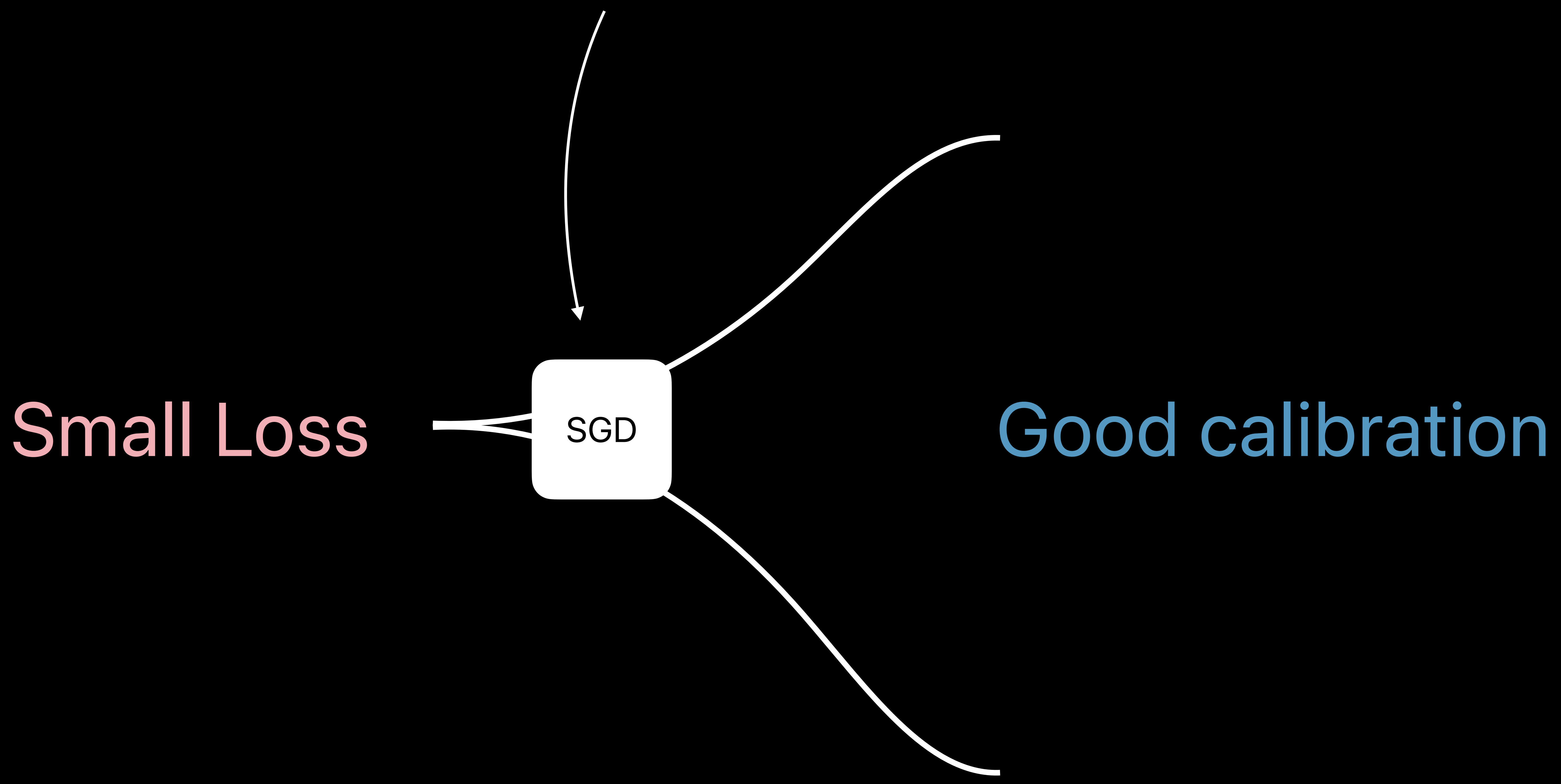
3. **(Theory assumption):**
   Structural risk minimization with any “well-behaved” complexity measure

$$\min_{f \in \mathcal{F}} \mathrm{MSE}_{\mathcal{D}}(f) + \lambda \mu(f).$$

Implications

- Generic characterization of when
  (sub-optimal) **loss-minimization** yields (near-optimal) **calibration**
- Importance of **depth** for calibration
- Importance of **proper scoring rules** for calibration
- Non-Baysean reasons for calibration

Q: What's important about this box?



A: Output is (nearly) post-processing-optimal w.r.t. loss

Thanks!

In Collaboration With

Parikshit Gopalan
Apple

Vimal Thilak
Apple

Omid Saremi
Apple

Joshua Suskind
Apple

Jarosław Błasiok
Columbia

Annabelle Carrell
Cambridge, Apple intern

Lunjia Hu
Stanford, Apple intern

Elan Rosenfeld
CMU, Apple intern

Defining “Almost All”

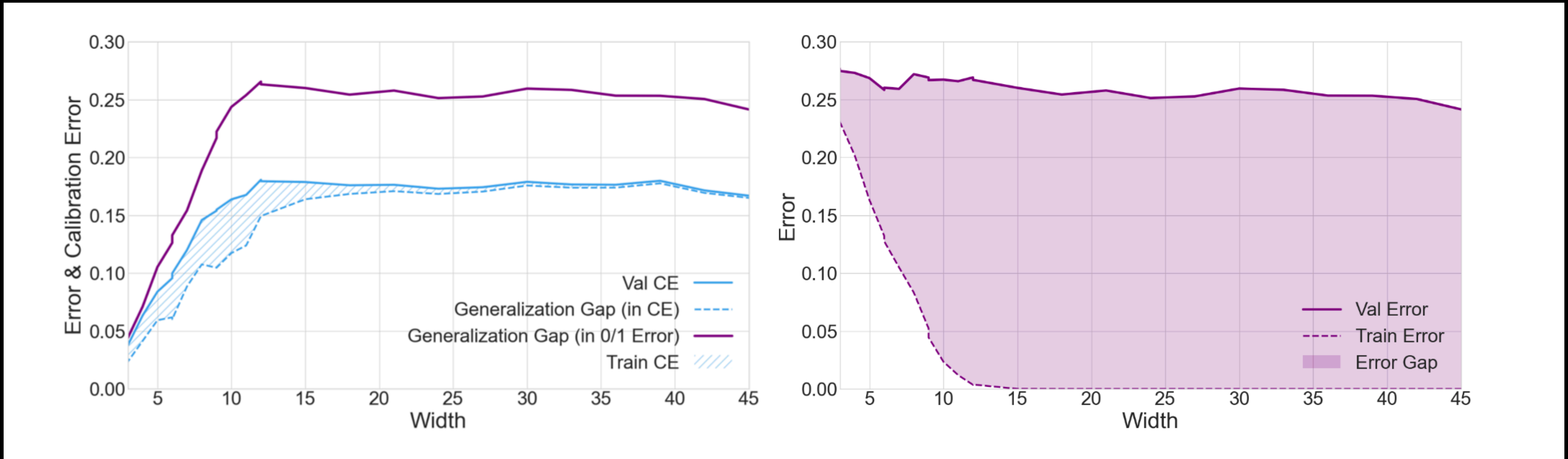
Definition-by-example:

Data distribution	Any
Architecture	Any* (MLP, ConvNet, Transformer,…)
Model depth	≥ 2
Model width	Any* (≥ 100)
Optimizer	Any* SGD-variant (SGD, Adam, …)
Optimization steps	Any* (≥ 10, after “warm-up” period)
Sample size	Any
Data-aug	None, or “standard” (measure-preserving)
Loss function	Any proper scoring rule (MSE, xent, …)
Regularization	None, or very weak (e.g. wd=1-e4)

Empirical Claim 1:

For almost all* ML models

$$\mu_{\text{Train}} \approx 0$$

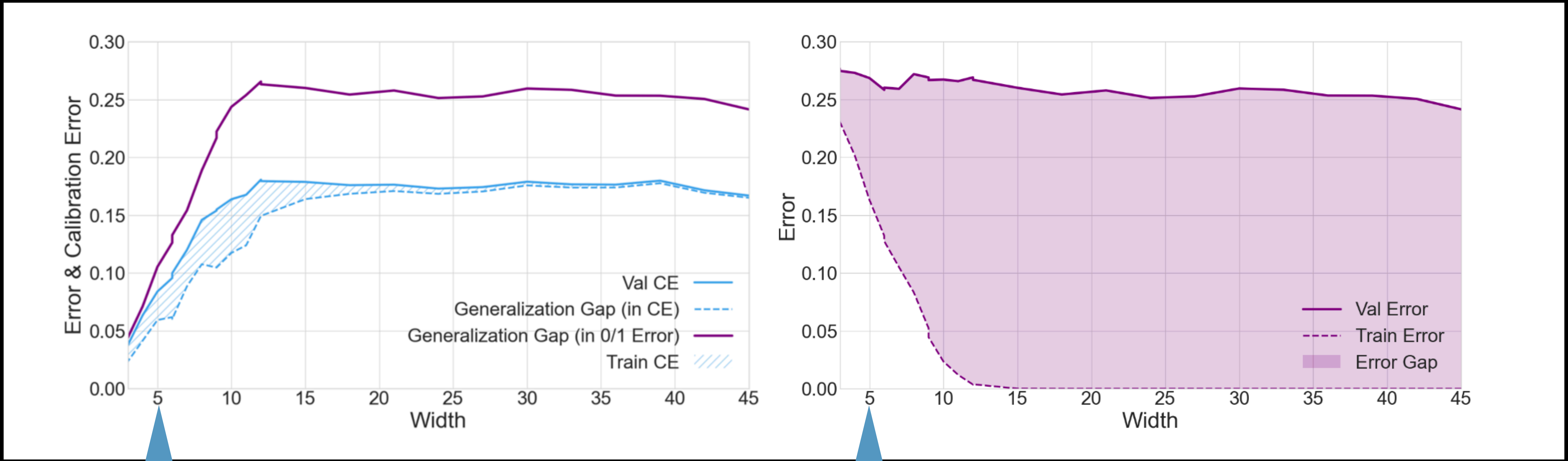


(ResNets on binary-CIFAR-10)

Empirical Claim 1:

For almost all* ML models

$$\mu_{\text{Train}} \approx 0$$



Surprising: small DNNs, with **high train error**, have good train calibration.

(ResNets on binary-CIFAR-10)

Part 1.
Measuring Miscalibration

Most models aren't *perfectly calibrated*.

How do we measure *degree-of-miscalibration*?

Summary: How to Measure Miscalibration [Błasiok, Gopalan, Hu, **N.** STOC 2023]

Most models aren't *perfectly calibrated*. How do we measure *degree-of-miscalibration*?

Desire:

Function $\mu_D(f) \in [0,\infty)$ that measures "degree of miscalibration"

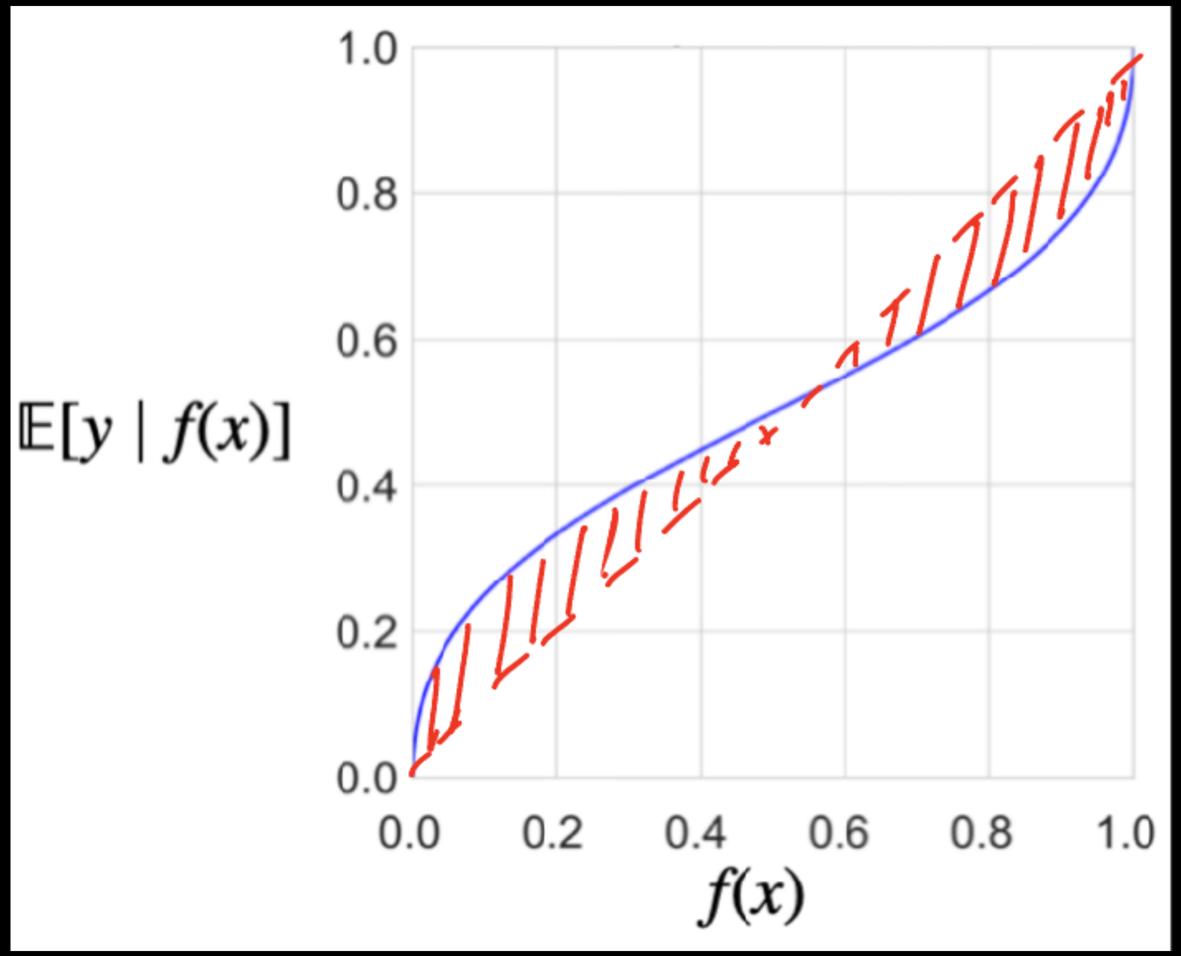
Summary: How to Measure Miscalibration [Błasiok, Gopalan, Hu, **N.** STOC 2023]

Most models aren't *perfectly calibrated*. How do we measure *degree-of-miscalibration*?

DON'T:

- Use "Expected Calibration Error (ECE)"

$$\mathrm{ECE}(f) = \mathbb{E}[\,|\,\mathbb{E}[y \mid f(x)] - f(x)\,|\,]$$



$\mathrm{ECE}(f)$ is discontinuous in f!

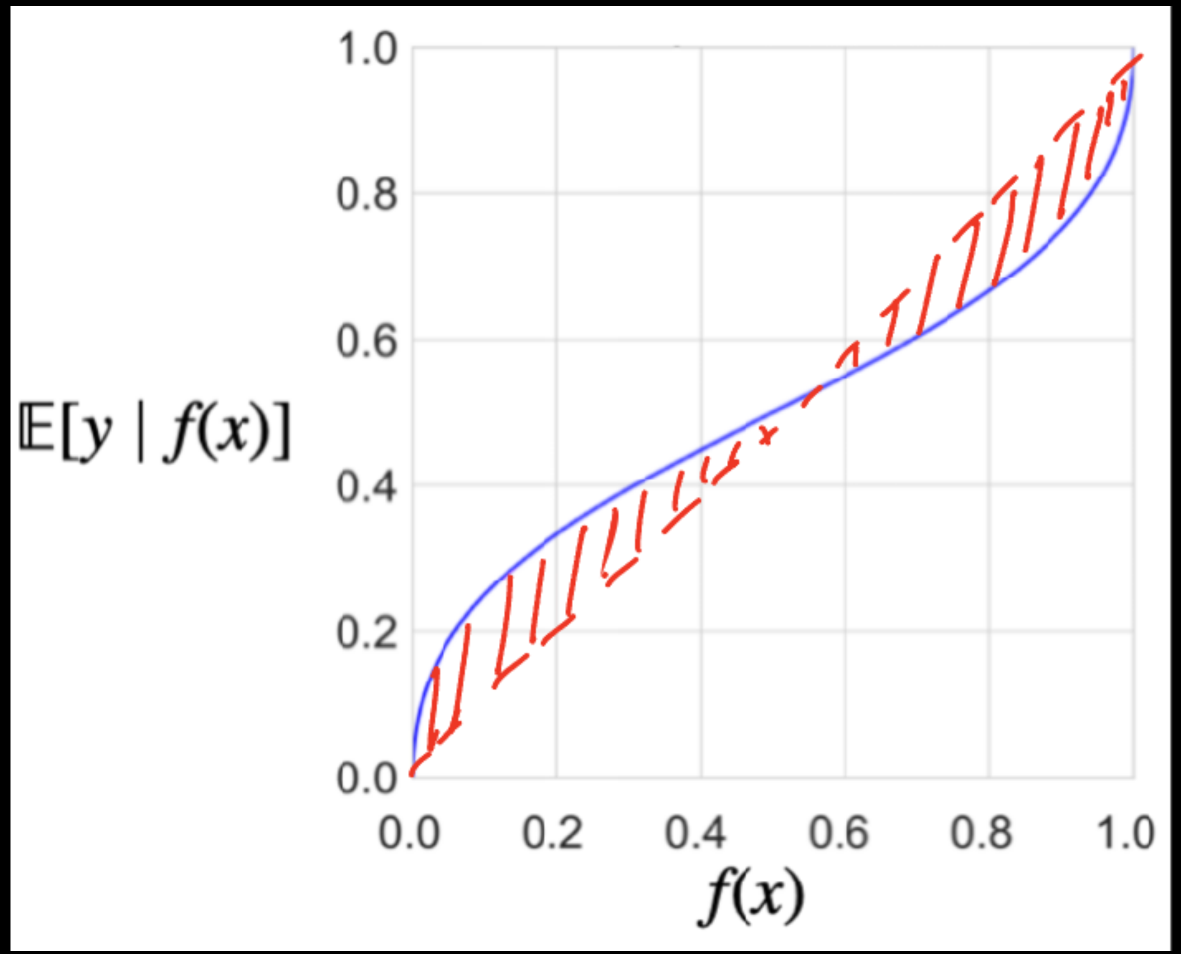
Summary: How to Measure Miscalibration [Błasiok, Gopalan, Hu, **N.** STOC 2023]

Most models aren't *perfectly calibrated*. How do we measure *degree-of-miscalibration*?

DON'T:

- Use "Expected Calibration Error (ECE)"

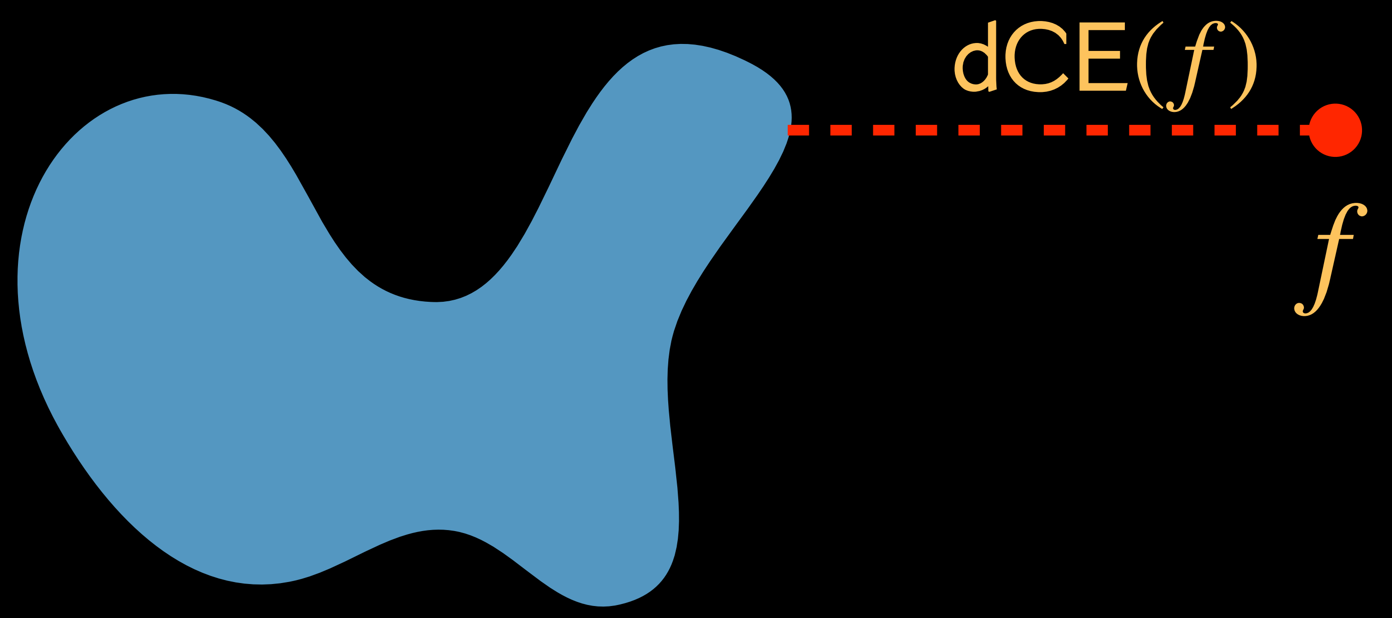
$$\mathrm{ECE}(f) = \mathbb{E}[\,|\,\mathbb{E}[y \mid f(x)] - f(x)\,|\,]$$



$\mathrm{ECE}(f)$ is discontinuous in f!

DO:

- Use "*ℓ_1 distance from perfect calibration*"



set of perfectly-calibrated functions

Summary: How to Measure Miscalibration

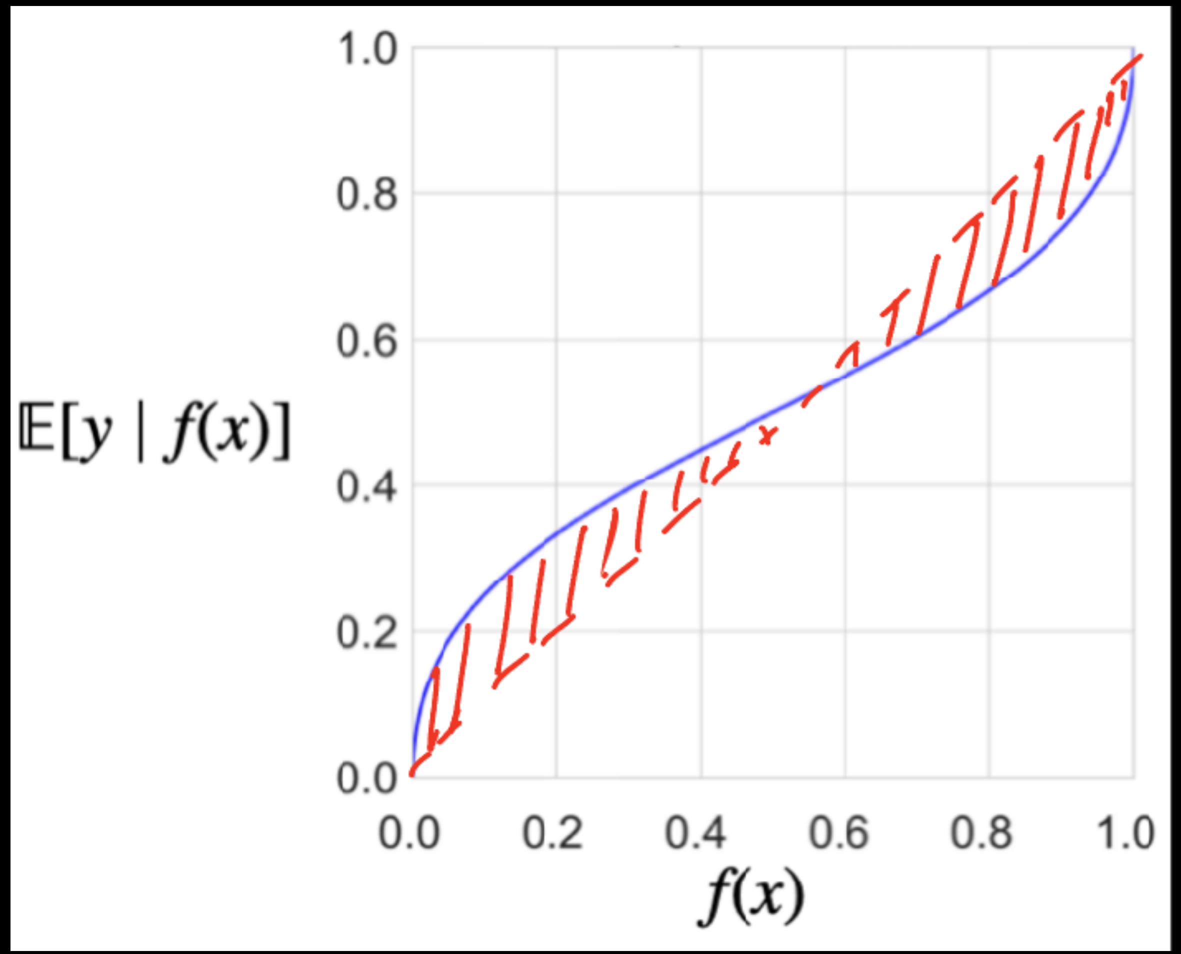
[Błasiok, Gopalan, Hu, **N.** STOC 2023]

Most models aren't *perfectly calibrated*. How do we measure *degree-of-miscalibration*?

DON'T:

- Use "Expected Calibration Error (ECE)"

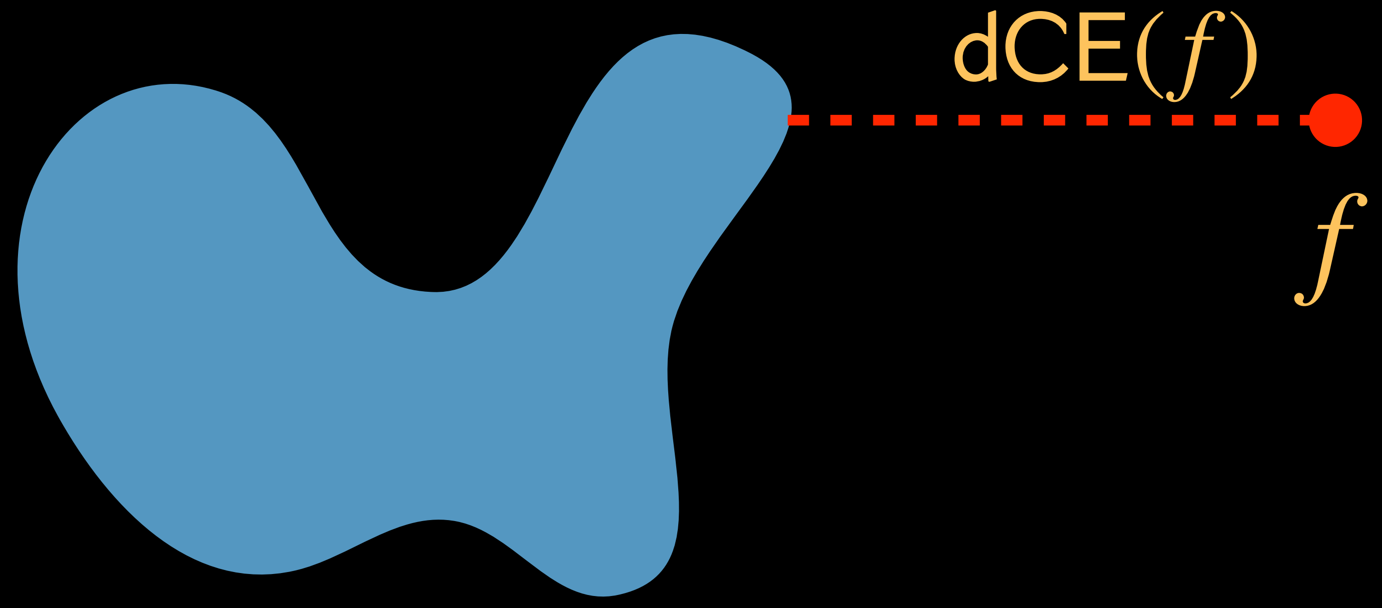
$$\mathrm{ECE}(f) = \mathbb{E}[\,|\,\mathbb{E}[y \mid f(x)] - f(x)\,|\,]$$



$\mathrm{ECE}(f)$ is discontinuous in f!

DO:

- Use "ℓ_1 *distance from perfect calibration*"



- Estimate with a "*consistent calibration metric*"
  e.g. Kernel calibration error (kCE)

$$\mathrm{kCE}_{\mathcal{D}}(f) := \sup_{w:\|w\|_K \le 1} \ \mathbb{E}_{(f,y)\sim\mathcal{D}_f} [w(f)(y - f)]$$

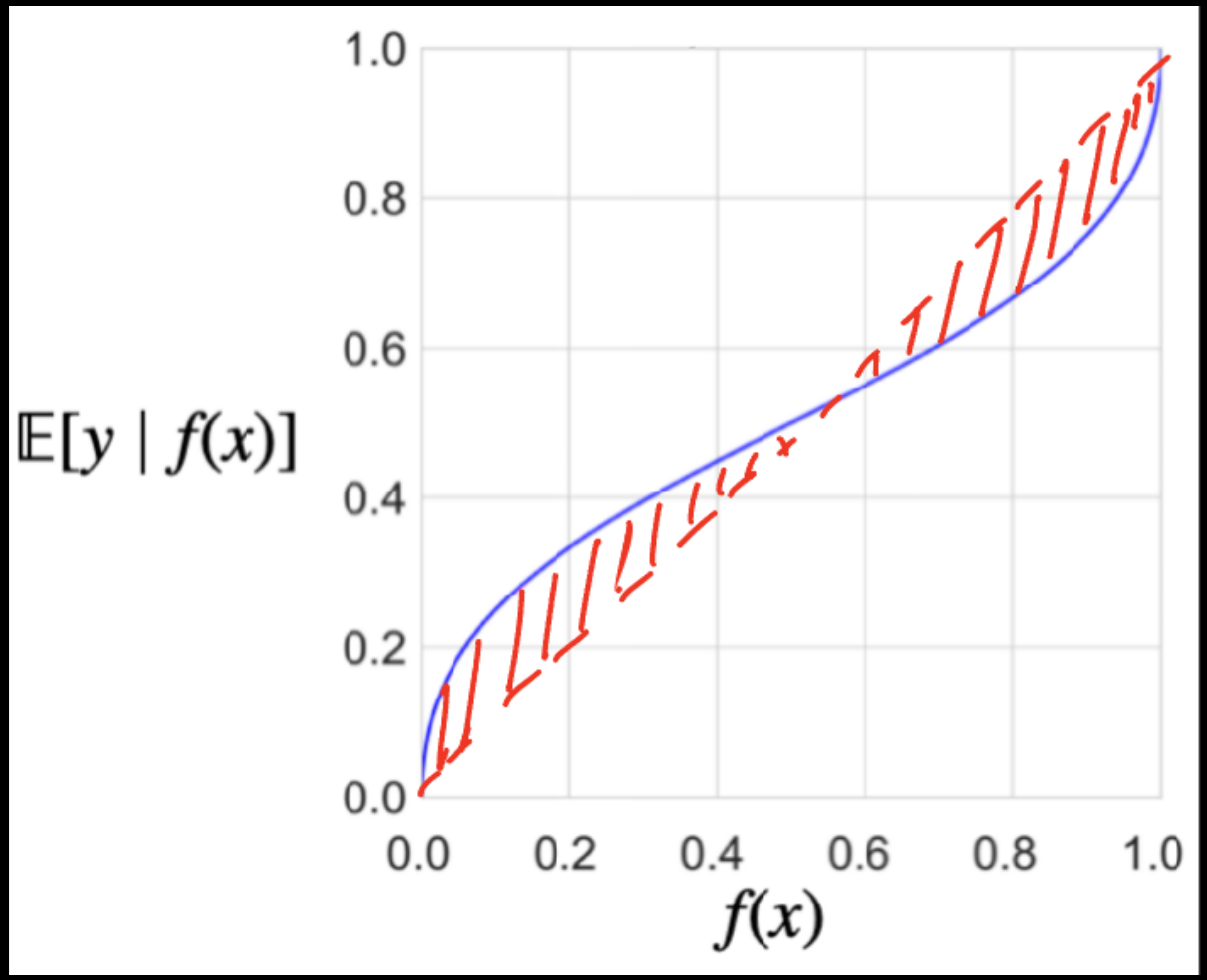
Measuring Miscalibration

Most models aren't *perfectly calibrated*.

How to measure **degree-of-miscalibration?**

Many proposed measures are problematic. Eg, ECE:

$$\mathrm{ECE}(f) = \mathbb{E}[\,|\,\mathbb{E}[y \mid f(x)] - f(x)\,|\,]$$



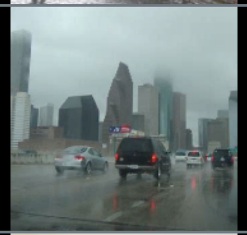
Problem: $\mathrm{ECE}(f)$ is discontinuous in f

1. $\|f_1 - f_2\| \leq \varepsilon$
2. $\mathrm{ECE}(f_1) - \mathrm{ECE}(f_2) \geq 0.5 - \varepsilon$

Problem: $\mathrm{ECE}(f)$ is discontinuous in f

1. $\|f_1 - f_2\| \leq \varepsilon$

2. $\mathrm{ECE}(f_1) - \mathrm{ECE}(f_2) \geq 0.5 - \varepsilon$

x	y	$f_1(x)$	$f_2(x)$
	1	0.5	0.5+ε
	1	0.5	0.5+ε
	1	0.5	0.5+ε
	0	0.5	0.5-ε
	0	0.5	0.5-ε
	0	0.5	0.5-ε

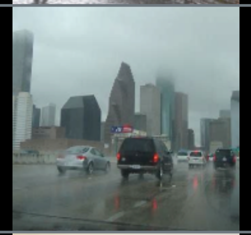
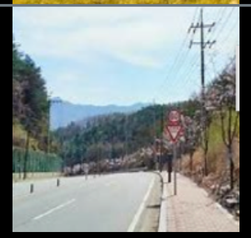
$\mathrm{ECE}(f_1) = 0$ $\qquad$ $\mathrm{ECE}(f_2) \approx 0.5$

$$\mathrm{ECE}(f) = \mathbb{E}[\,|\,\mathbb{E}[y \mid f(x)] - f(x)\,|\,]$$

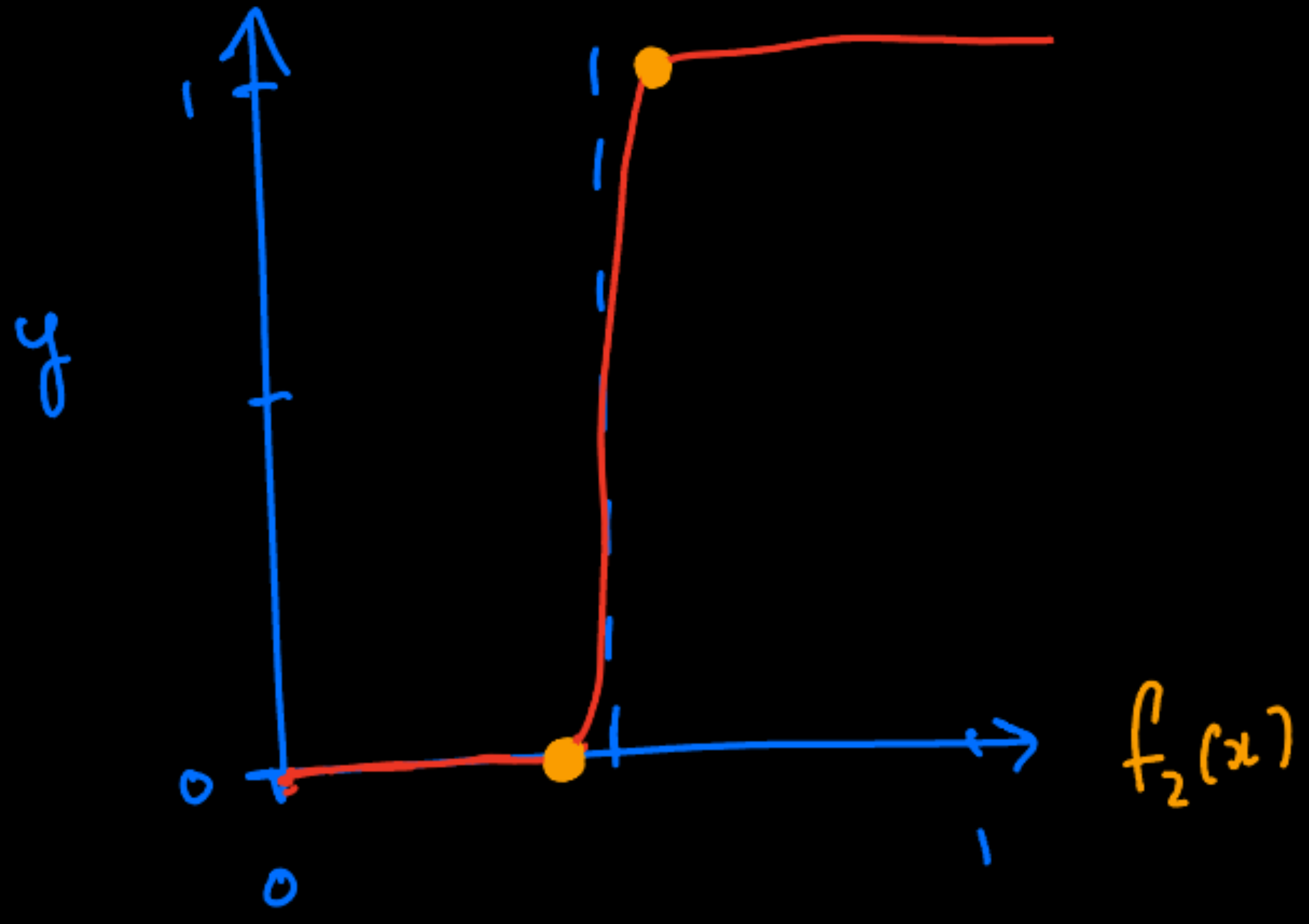
Problem: $\mathrm{ECE}(f)$ is discontinuous in f

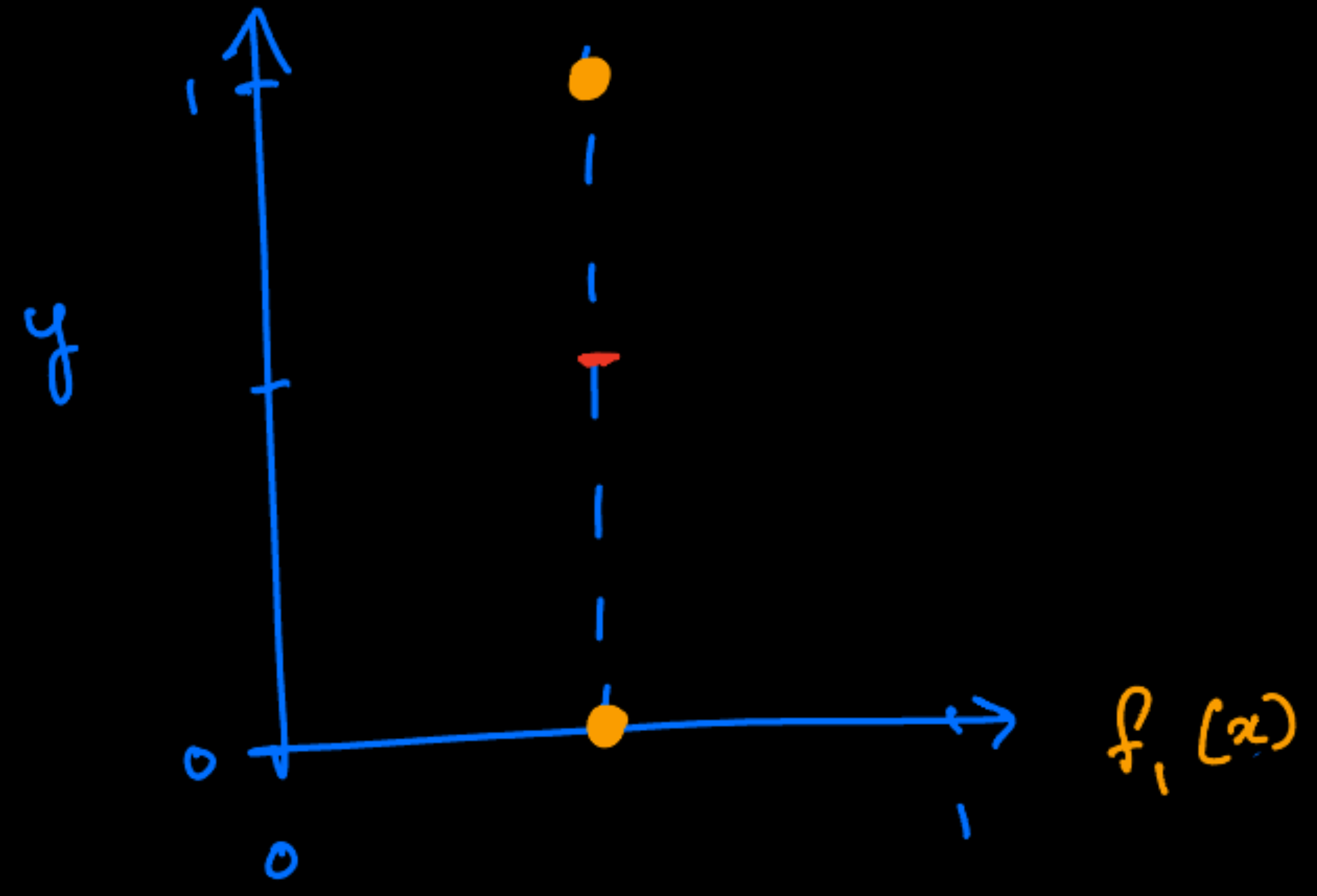
1. $\|f_1 - f_2\| \leq \varepsilon$

2. $\mathrm{ECE}(f_1) - \mathrm{ECE}(f_2) \geq 0.5 - \varepsilon$

x	y	$f_1(x)$	$f_2(x)$
	1	0.5	0.5+ε
	1	0.5	0.5+ε
	1	0.5	0.5+ε
	0	0.5	0.5-ε
	0	0.5	0.5-ε
	0	0.5	0.5-ε

$\mathrm{ECE}(f_1) = 0$ $\quad$ $\mathrm{ECE}(f_2) \approx 0.5$





$$\mathrm{ECE}(f) = \mathbb{E}[\,|\,\mathbb{E}[y \mid f(x)] - f(x)\,|\,]$$

Axiomatic construction of **degree-of-miscalibration** $\mu_D(f)$ **?**

Want $\mu(f) \in \mathbb{R}_{\geq 0}$ to satisfy:

1. Correctness:

$\mu(f) = 0 \iff f$ is perfectly calibrated

2. $\mu(f)$ is continuous in f

3. Can be estimated from samples

Axiomatic construction of **degree-of-miscalibration** $\mu_D(f)$ **?**

Want $\mu(f) \in \mathbb{R}_{\geq 0}$ to satisfy:

1. Correctness:

$$\mu(f) = 0 \iff f \text{ is perfectly calibrated}$$

2. $\mu(f)$ is continuous in f

3. Can be estimated from samples

	Correctness	Continuity	Estimation
ECE	✅	❌	❌
Binned-ECE	❌	❌	✅
Brier	❌	✅	✅
NLL	❌	✅	✅
NCE	❌	✅	✅
kCE/MMCE	✅	✅	✅
smCE	✅	✅	✅

Correctness:

$\mu(f) = 0 \iff f$ is perfectly calibrated

Robust Correctness (informally):

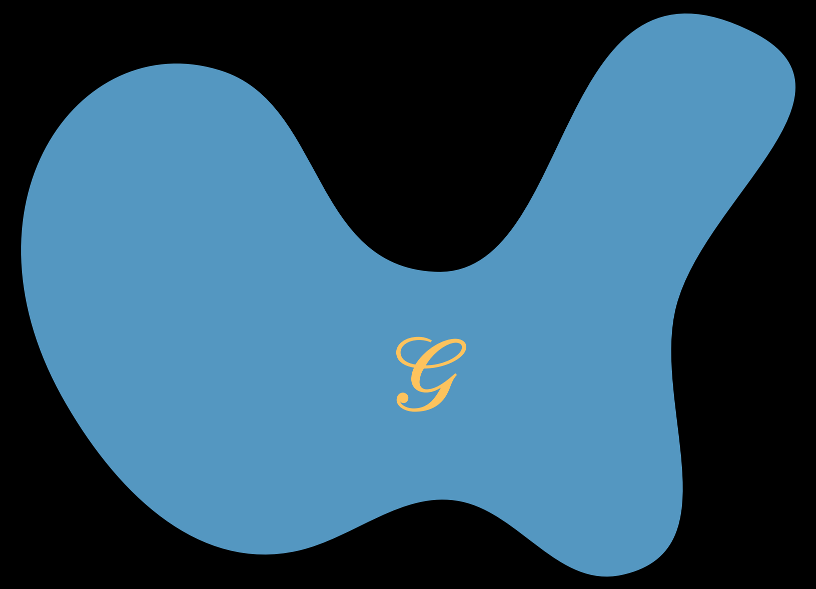
$\mu(f)$ is "close" to 0 $\iff$ f is "close" to perfectly calibrated

Correctness:

$\mu(f) = 0 \iff f$ is perfectly calibrated

Robust Correctness (informally):

$\mu(f)$ is “close” to 0 $\iff f$ is “close” to perfectly calibrated



$\mathscr{G}$:= set of perfectly-calibrated functions

Correctness:

$\mu(f) = 0 \iff f$ is perfectly calibrated

Robust Correctness (informally):

$\mu(f)$ is “close” to 0 $\iff f$ is “close” to perfectly calibrated

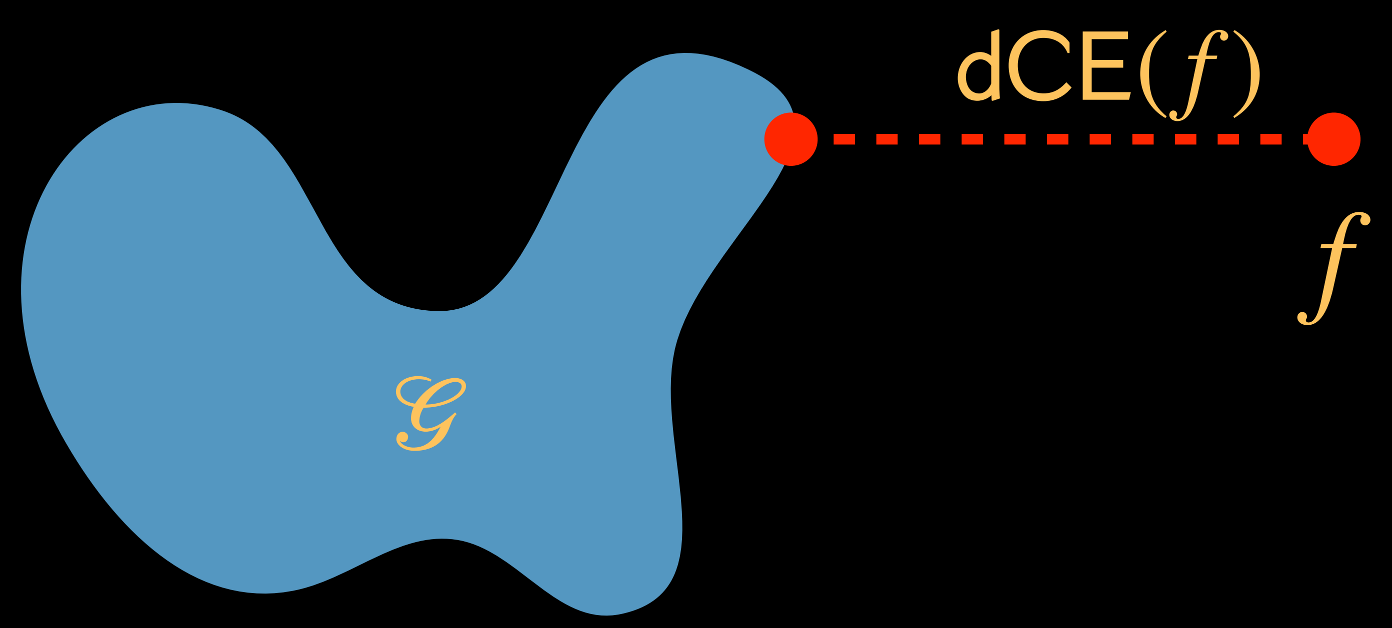
$\mathscr{G}$:= set of perfectly-calibrated functions

Correctness:

$\mu(f) = 0 \iff f$ is perfectly calibrated

Robust Correctness (informally):

$\mu(f)$ is "close" to 0 $\iff$ f is "close" to perfectly calibrated



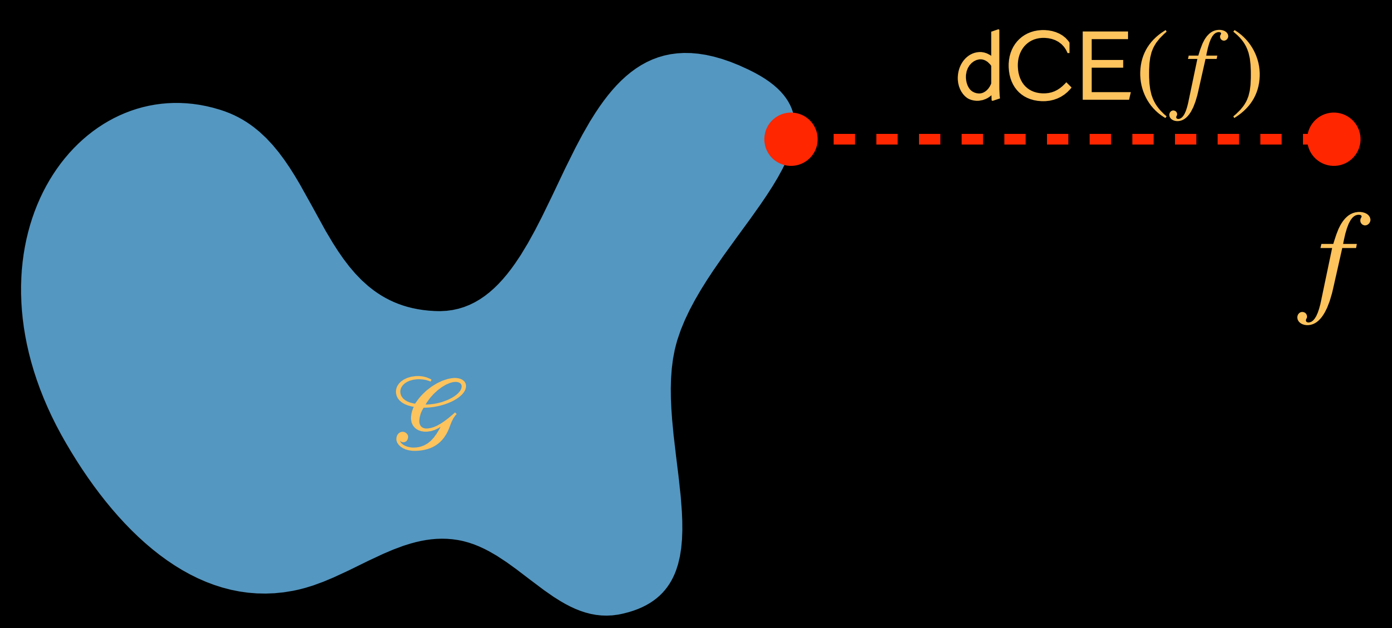
$\mathscr{G}$:= set of perfectly-calibrated functions

Correctness:

$\mu(f) = 0 \iff f$ is perfectly calibrated

Robust Correctness (informally):

$\mu(f)$ is "close" to 0 $\iff f$ is "close" to perfectly calibrated



$\mathscr{G}$:= set of perfectly-calibrated functions

The Calibration Distance

$$\mathrm{dCE}(f) := \min_{g \in \mathscr{G}} d_1(f, g)$$

Correctness:

$\mu(f) = 0 \iff f$ is perfectly calibrated

Robust Correctness (informally):

$\mu(f)$ is "close" to 0 $\iff f$ is "close" to perfectly calibrated

Robust Correctness:

$\mathrm{dCE}(f)^{\beta} \leq \mu(f) \leq \mathrm{dCE}(f)^{\alpha}$

The Calibration Distance

$\mathrm{dCE}(f) := \min_{g \in \mathcal{G}} d_1(f, g)$

Why not use $\mathrm{dCE}(f)$ as calibration measure μ?

Satisfies robust completeness:

$\mu(f)$ is "close" to 0 $\iff$ f is "close" to perfectly calibrated

The Calibration Distance

$$\mathrm{dCE}(f) := \min_{g \in \mathcal{G}} d_1(f, g)$$

Why not use $\mathrm{dCE}(f)$ as calibration measure μ?

Satisfies robust completeness:

$\mu(f)$ is "close" to 0 $\iff$ f is "close" to perfectly calibrated

Q: How to estimate from samples $\{(f(x_i), y_i)\}$?

- Both info-theoretic, and computational issues...

The Calibration Distance

$$\mathrm{dCE}(f) := \min_{g \in \mathcal{G}} d_1(f, g)$$

Unification

New metric $\mathsf{dCE}(f)$ intimately related to existing metrics:

- $\mathsf{kCE}(f)$: kernel calibration / MMCE [Kumar Sarawagi Jain 2018]
- $\mathsf{smCE}(f)$: smooth calibration [Foster Hart 2018]
- $\mathsf{intCE}(f)$: interval calibration

Unification

New metric $\mathrm{dCE}(f)$ intimately related to existing metrics:

- $\mathrm{kCE}(f)$: kernel calibration / MMCE [Kumar Sarawagi Jain 2018]
- $\mathrm{smCE}(f)$: smooth calibration [Foster Hart 2018]
- $\mathrm{intCE}(f)$: interval calibration

Theorem: For $\mu \in \{\mathrm{kCE}, \mathrm{smCE}, \mathrm{intCE}\}$:

$$\mu^2 \leq \mathrm{dCE} \leq \mu^{1/3}$$

Unification

New metric $\mathsf{dCE}(f)$ intimately related to existing metrics:

- $\mathsf{kCE}(f)$: kernel calibration / MMCE [Kumar Sarawagi Jain 2018]
- $\mathsf{smCE}(f)$: smooth calibration [Foster Hart 2018]
- $\mathsf{intCE}(f)$: interval calibration

Theorem: For $\mu \in \{\mathsf{kCE}, \mathsf{smCE}, \mathsf{intCE}\}$:

$$\mu^2 \leq \mathsf{dCE} \leq \mu^{1/3}$$

Takeaway:

1. Estimate dCE from samples
2. Prior metrics are related

Practical Takeaways

Measure calibration with either:

1. Kernel Calibration Error

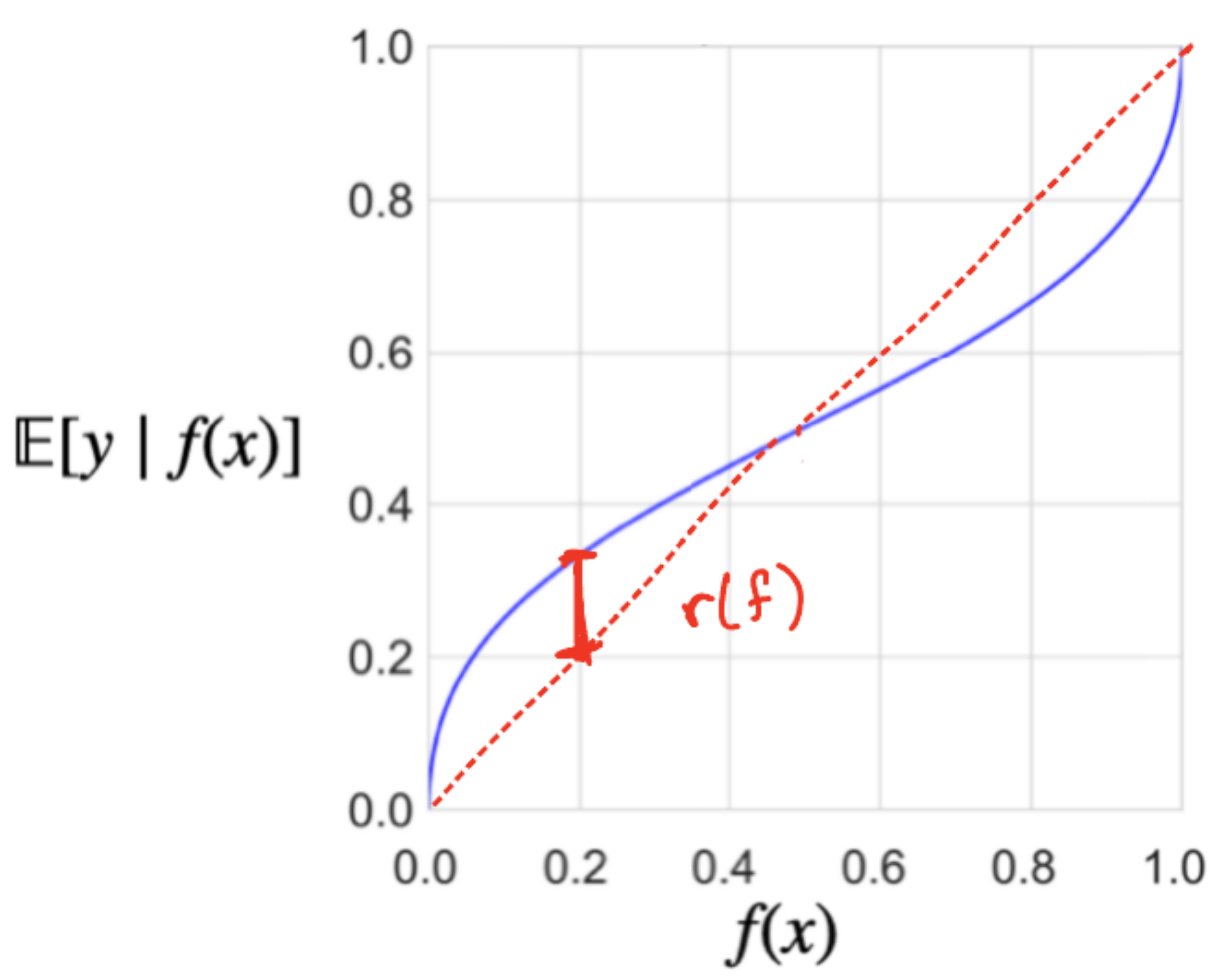
2. Interval Calibration Error (modification of binnedECE)

Kernel Calibration Error

$$\mathrm{ECE}_{\mathcal{D}}(f) := \sup_{w:[0,1]\to[-1,1]} \mathop{\mathbb{E}}_{(f,y)\sim\mathcal{D}_f} [w(f)(y - f)]$$

Kernel Calibration Error

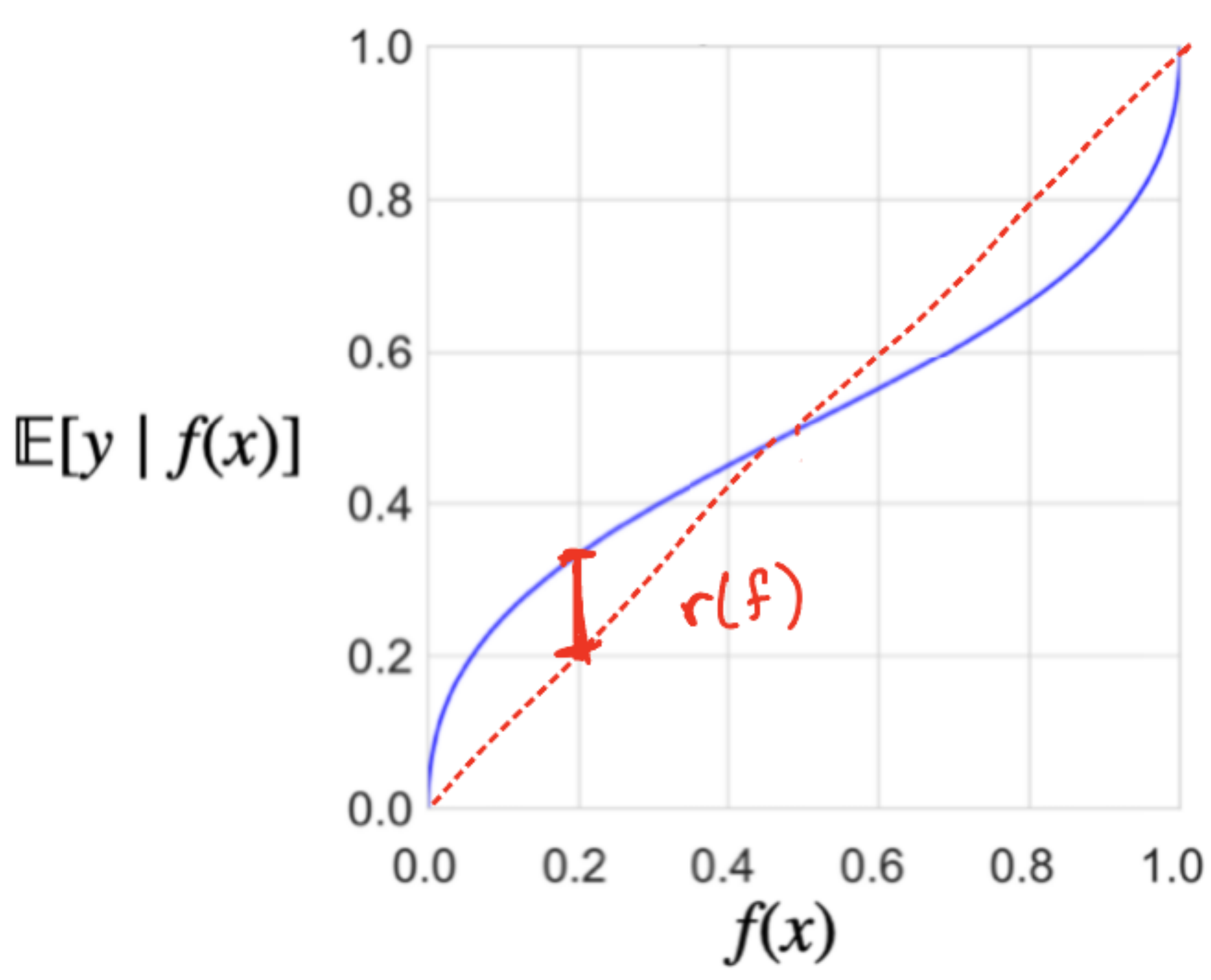
$$\mathrm{ECE}_{\mathcal{D}}(f) := \sup_{w:[0,1]\to[-1,1]} \mathop{\mathbb{E}}_{(f,y)\sim\mathcal{D}_f} [w(f)(y-f)]$$

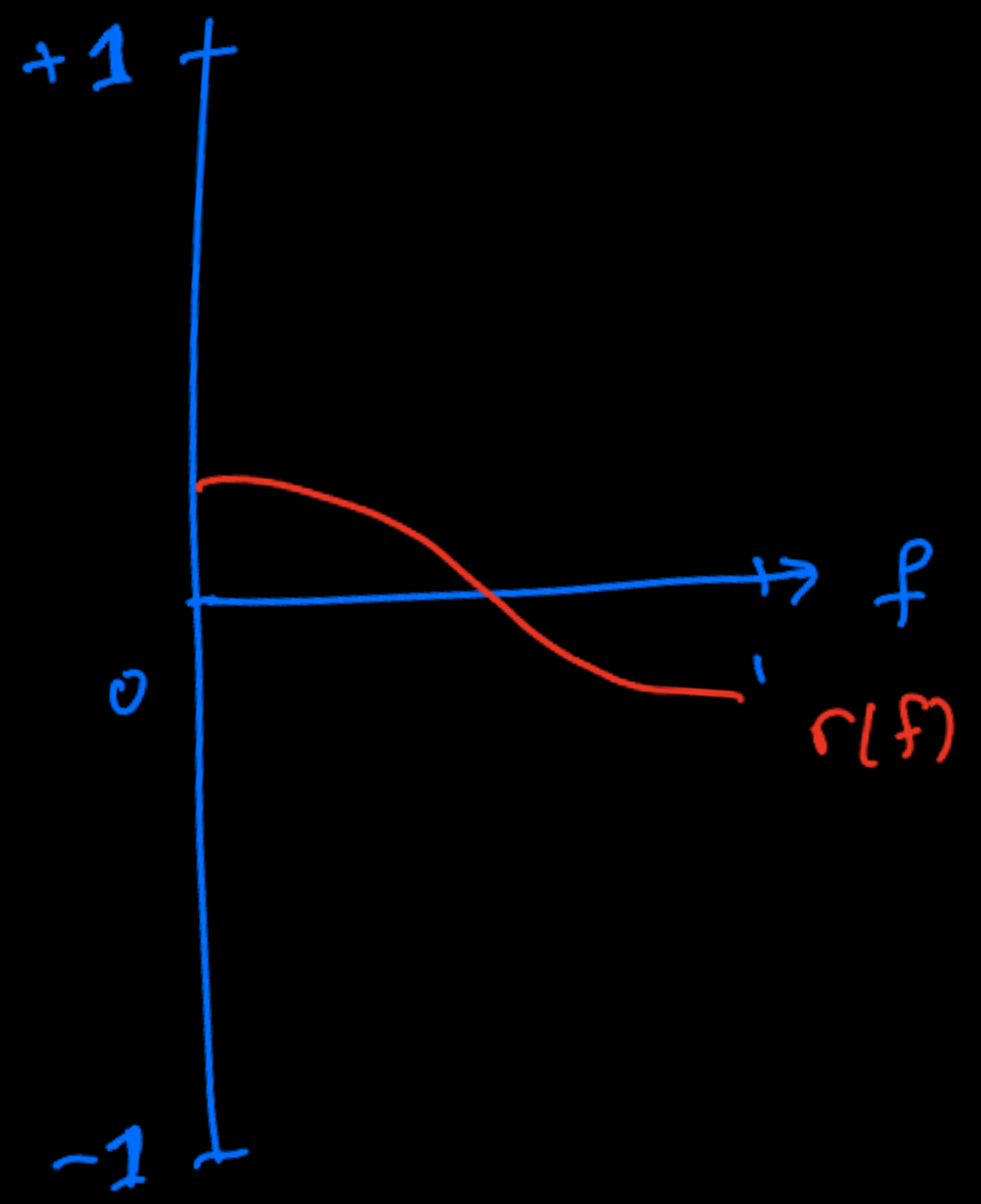


"Residual": $r(f) := \mathbb{E}[y \mid f] - f$

Kernel Calibration Error

$$\mathrm{ECE}_{\mathcal{D}}(f) := \sup_{w:[0,1]\to[-1,1]} \mathop{\mathbb{E}}_{(f,y)\sim\mathcal{D}_f} [w(f)(y-f)]$$

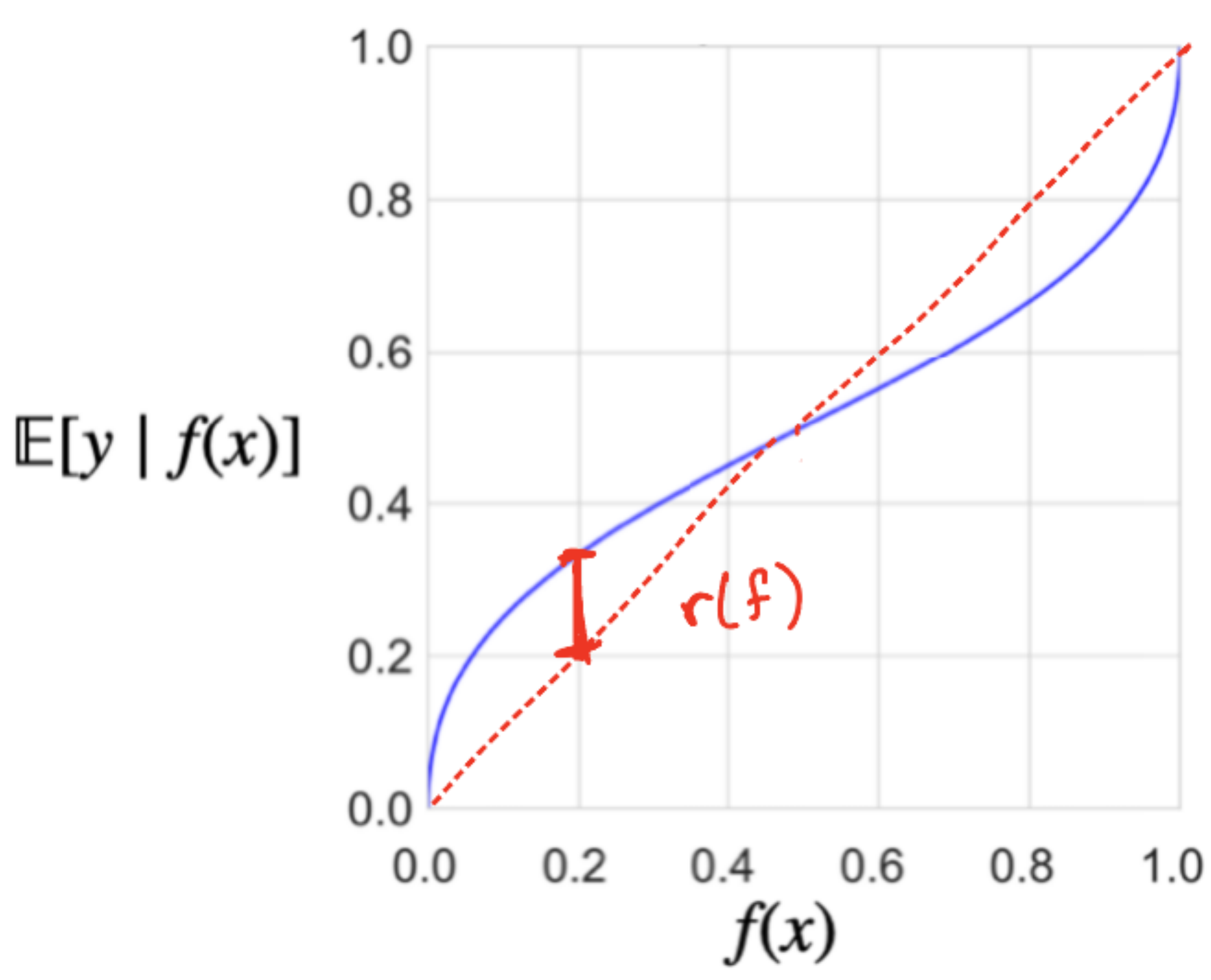


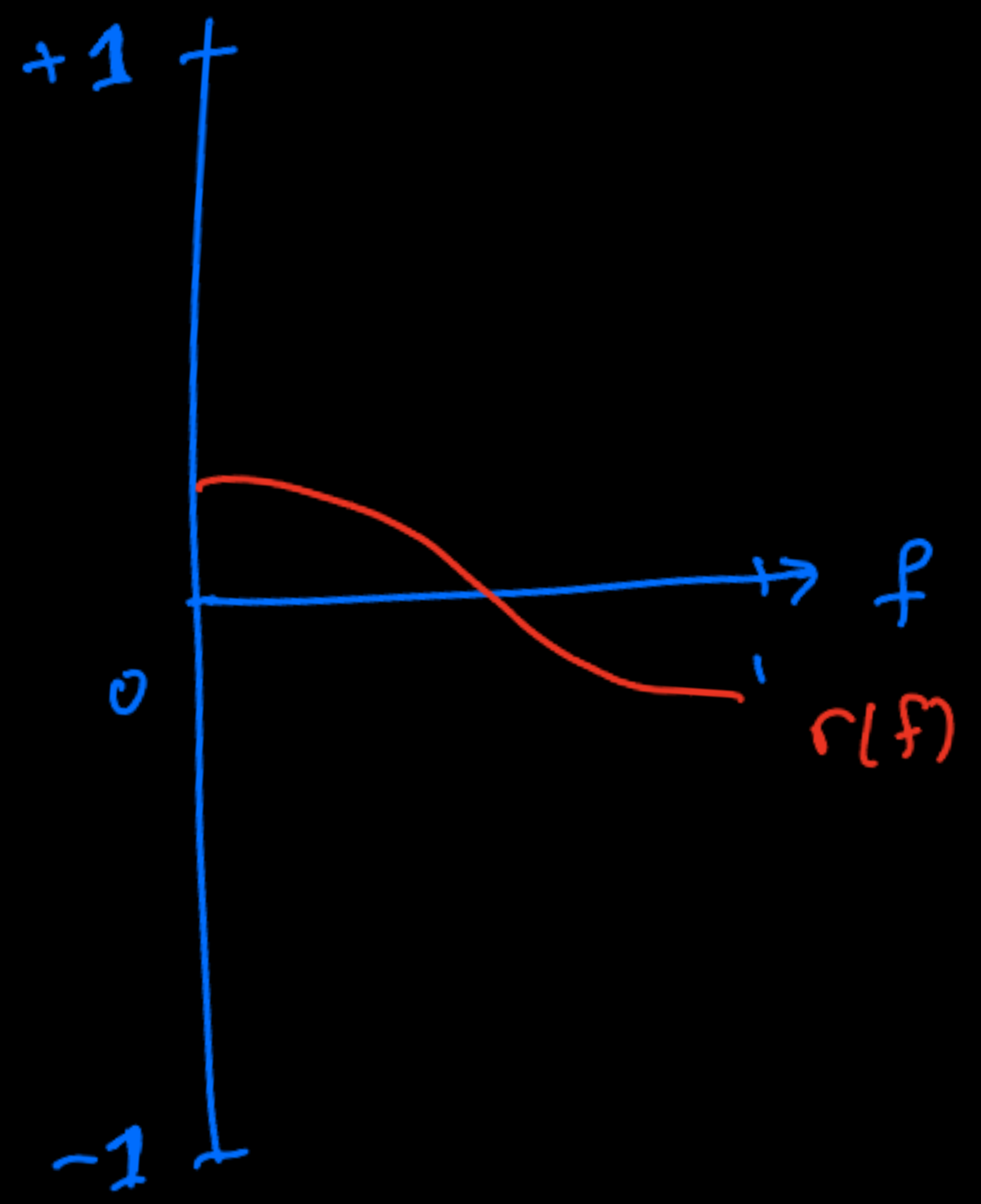


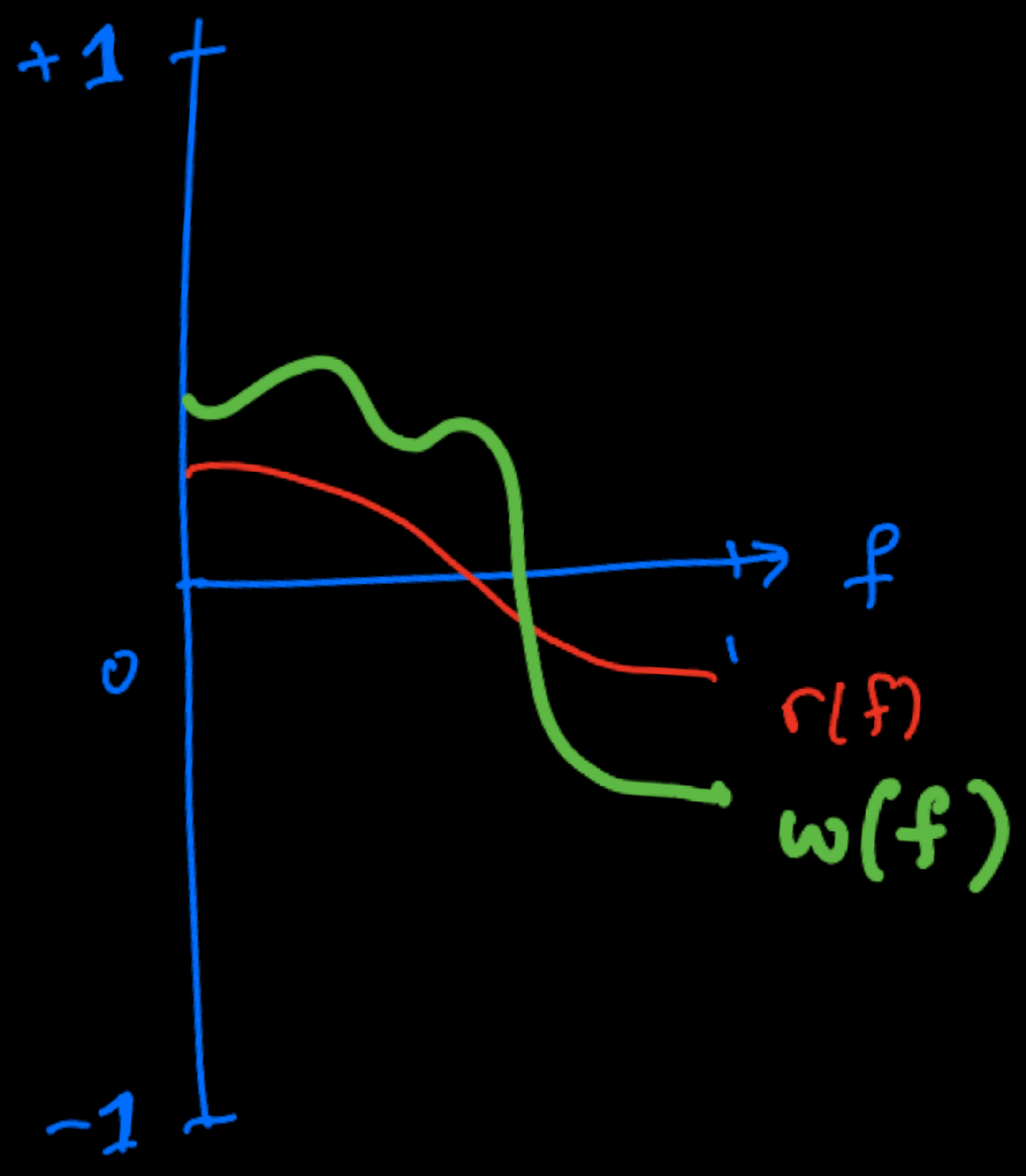
"Residual": $r(f) := \mathbb{E}[y \mid f] - f$

Kernel Calibration Error

$$\mathrm{ECE}_{\mathcal{D}}(f) := \sup_{w:[0,1]\to[-1,1]} \mathop{\mathbb{E}}_{(f,y)\sim\mathcal{D}_f} [w(f)(y-f)]$$





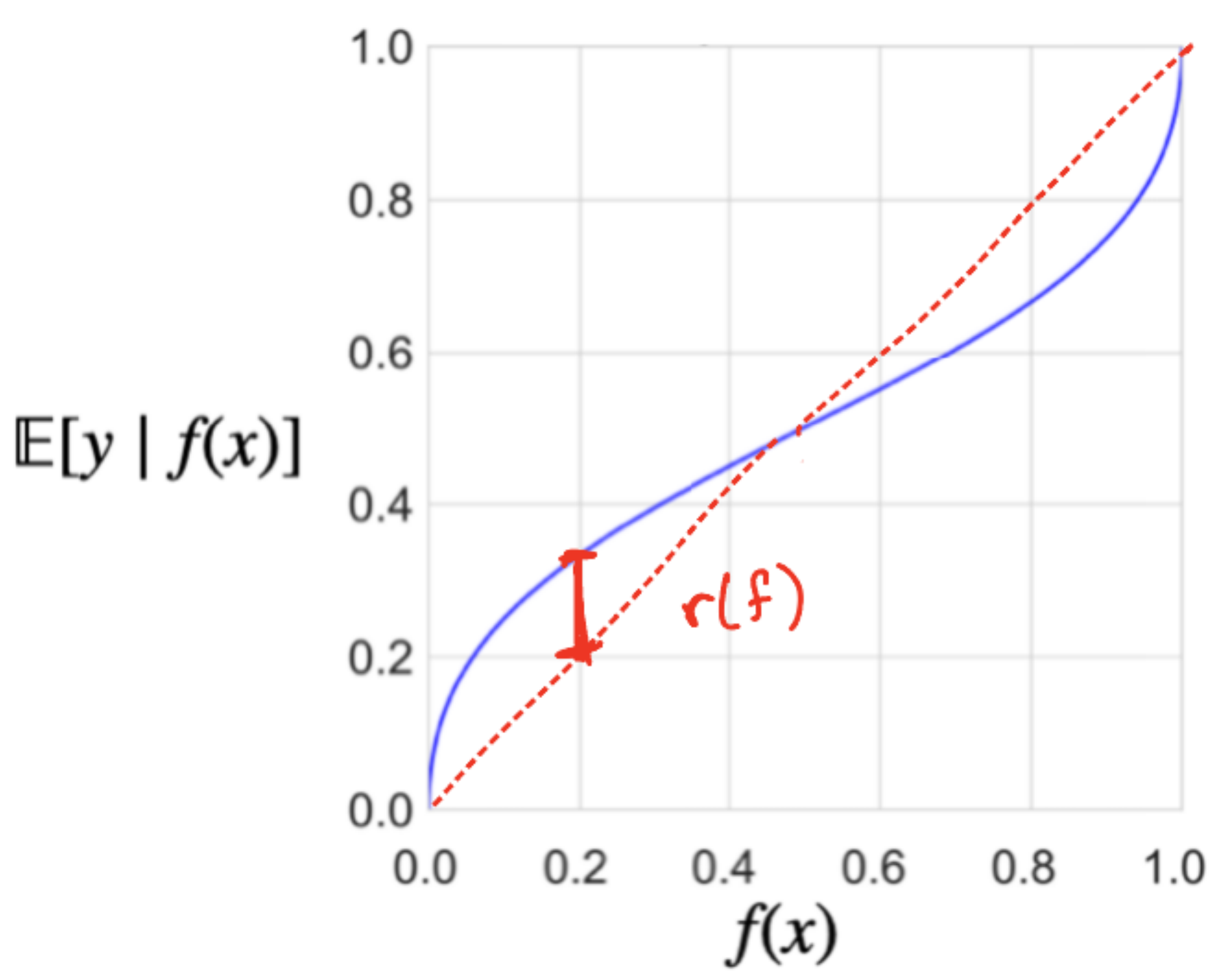


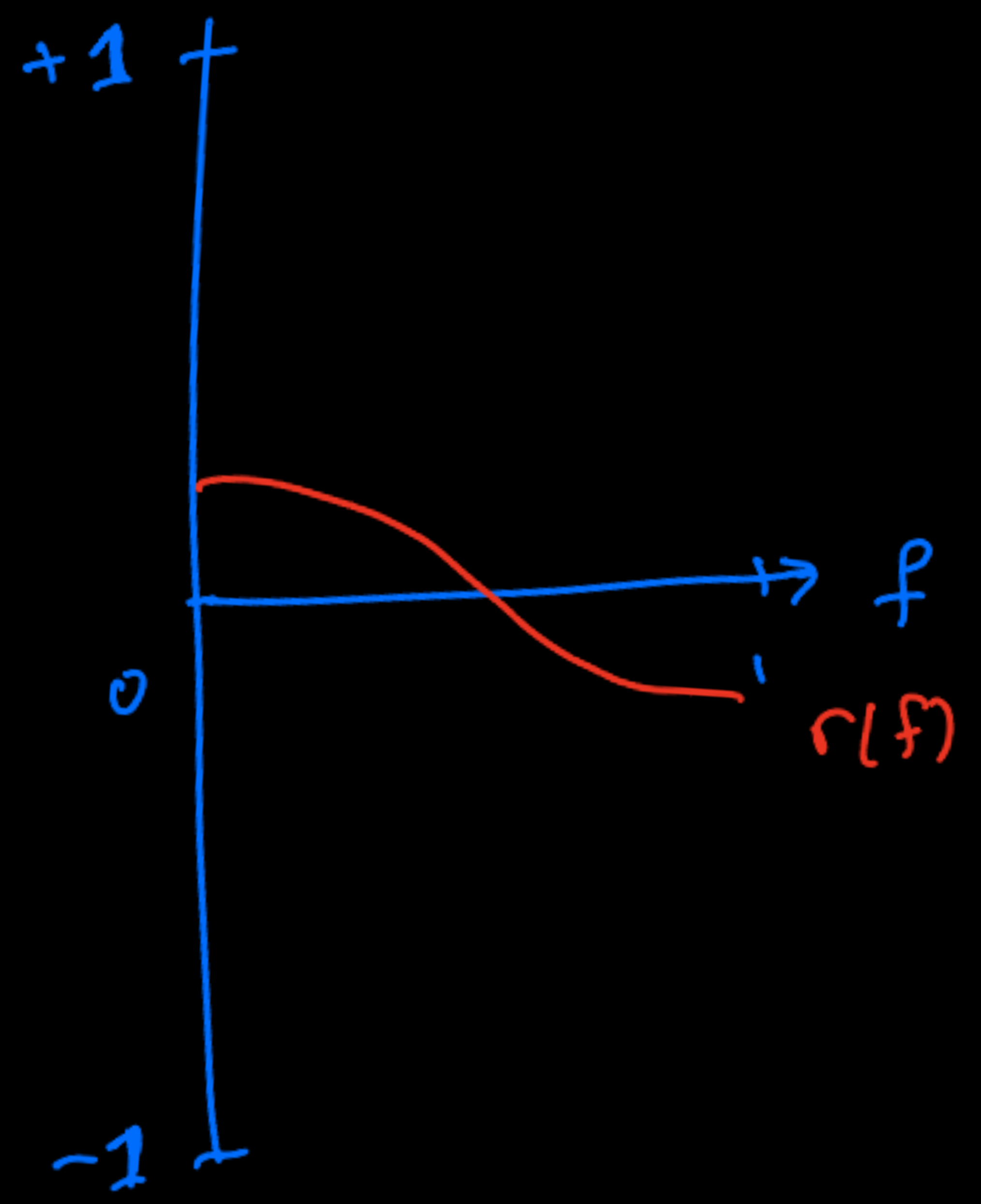
“Residual”: $r(f) := \mathbb{E}[y \mid f] - f$

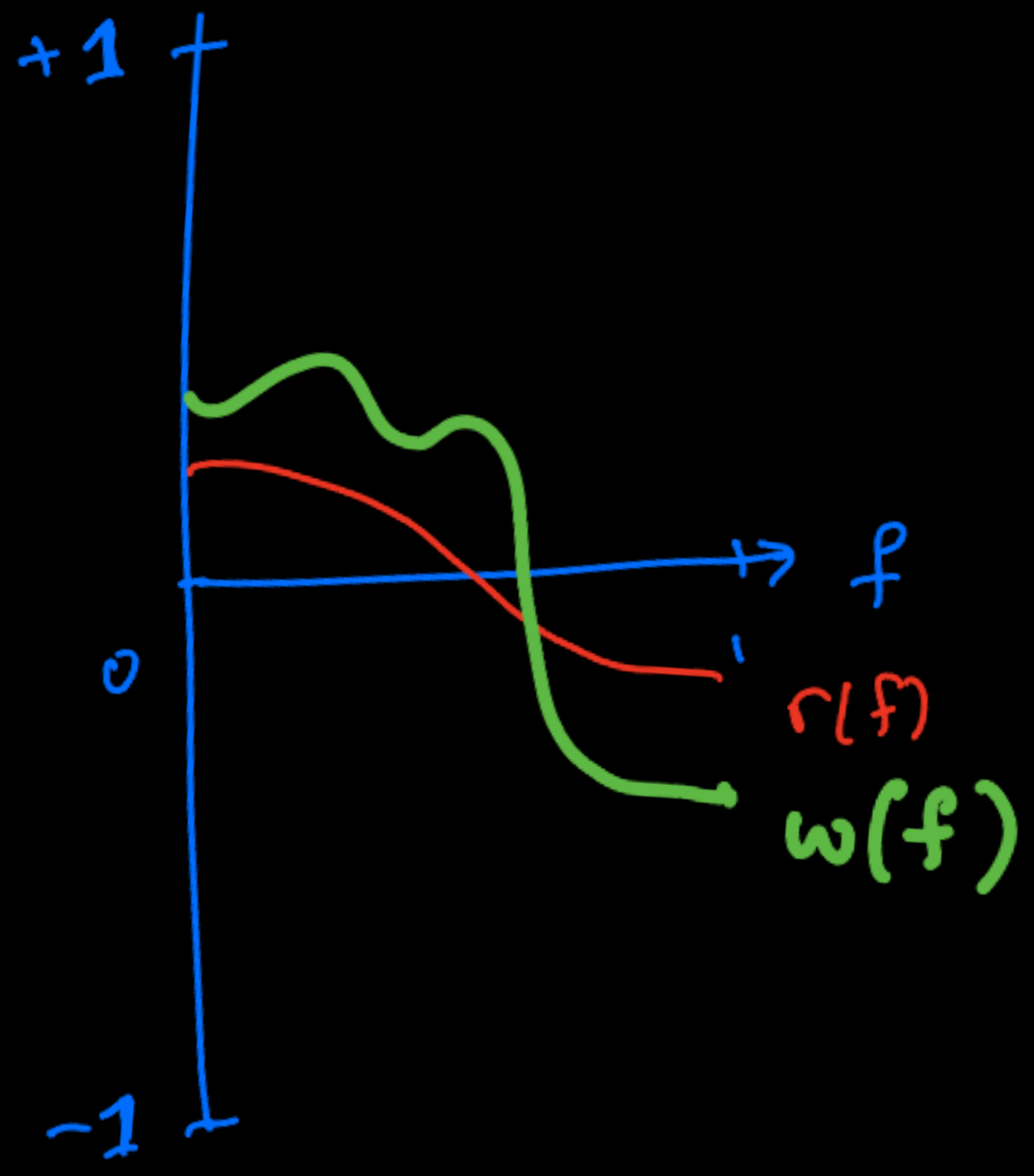
Kernel Calibration Error

$$\mathrm{ECE}_{\mathcal{D}}(f) := \sup_{w:[0,1]\to[-1,1]} \mathop{\mathbb{E}}_{(f,y)\sim\mathcal{D}_f} [w(f)(y-f)]$$

$$\mathrm{kCE}_{\mathcal{D}}(f) := \sup_{w:\|w\|_K\leq 1} \mathop{\mathbb{E}}_{(f,y)\sim\mathcal{D}_f} [w(f)(y-f)]$$







"Residual": $r(f) := \mathbb{E}[y \mid f] - f$

Kernel Calibration Error: Sample Estimation

Given: Samples $(f(x_i), y_i) =: (f_i, y_i)$

Residuals: (f_i, r_i) for $r_i := (y_i - f_i)$

Kernel Calibration Error: Sample Estimation

Given: Samples $(f(x_i), y_i) =: (f_i, y_i)$

Residuals: (f_i, r_i) for $r_i := (y_i - f_i)$

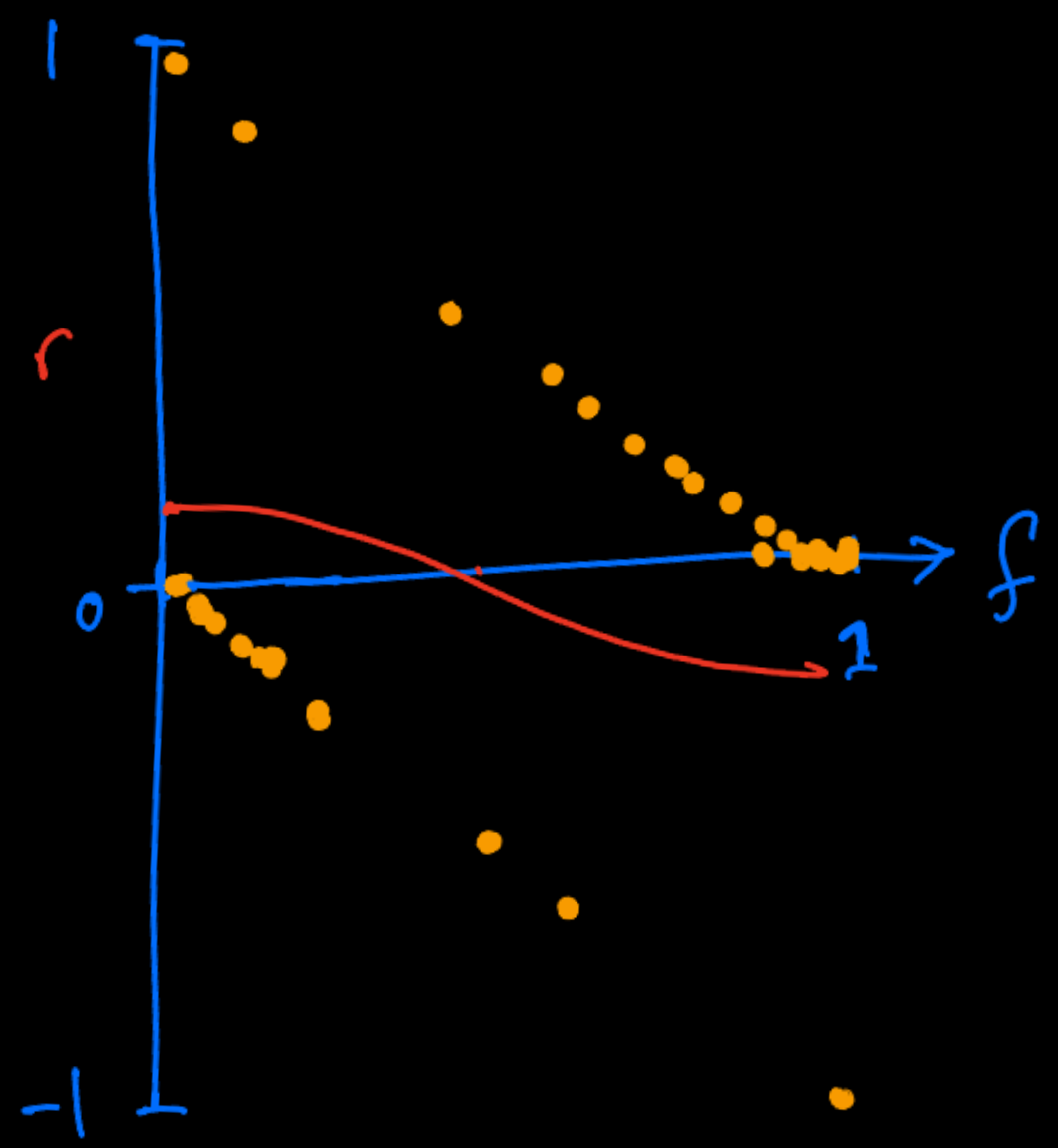
$$\widehat{\mathrm{kCE}_{\mathcal{D}}}(f) = \sqrt{\frac{1}{n^2}\sum_{i,j} r_i r_j K(f_i, f_j)} = \|r\|_{K(f,f)}$$

Kernel Calibration Error: Sample Estimation

Given: Samples $(f(x_i), y_i) =: (f_i, y_i)$

Residuals: (f_i, r_i) for $r_i := (y_i - f_i)$

$$\widehat{\mathrm{kCE}_{\mathcal{D}}}(f) = \sqrt{\frac{1}{n^2}\sum_{i,j} r_i r_j K(f_i, f_j)} = \|r\|_{K(f,f)}$$



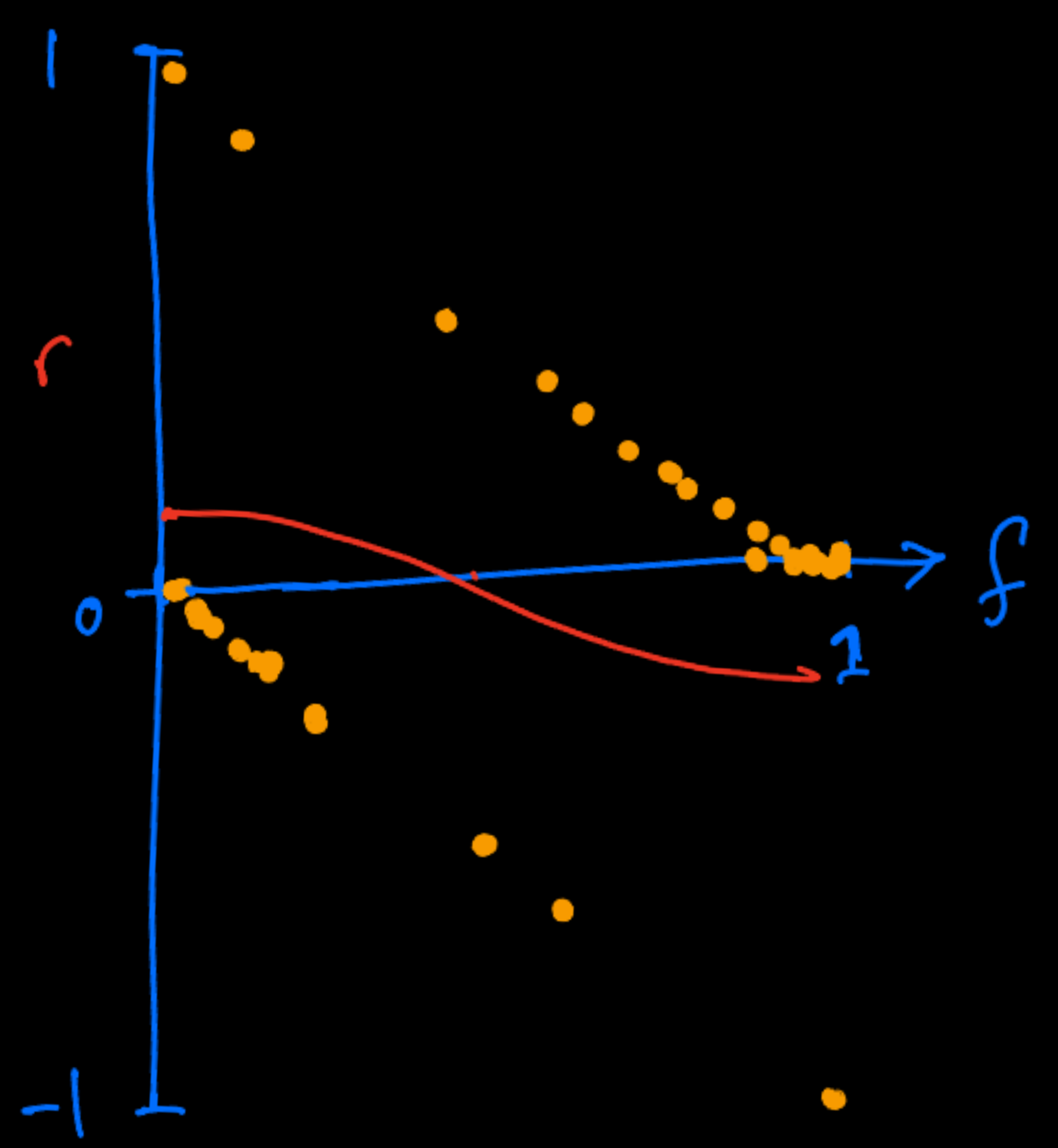
Kernel Calibration Error: Sample Estimation

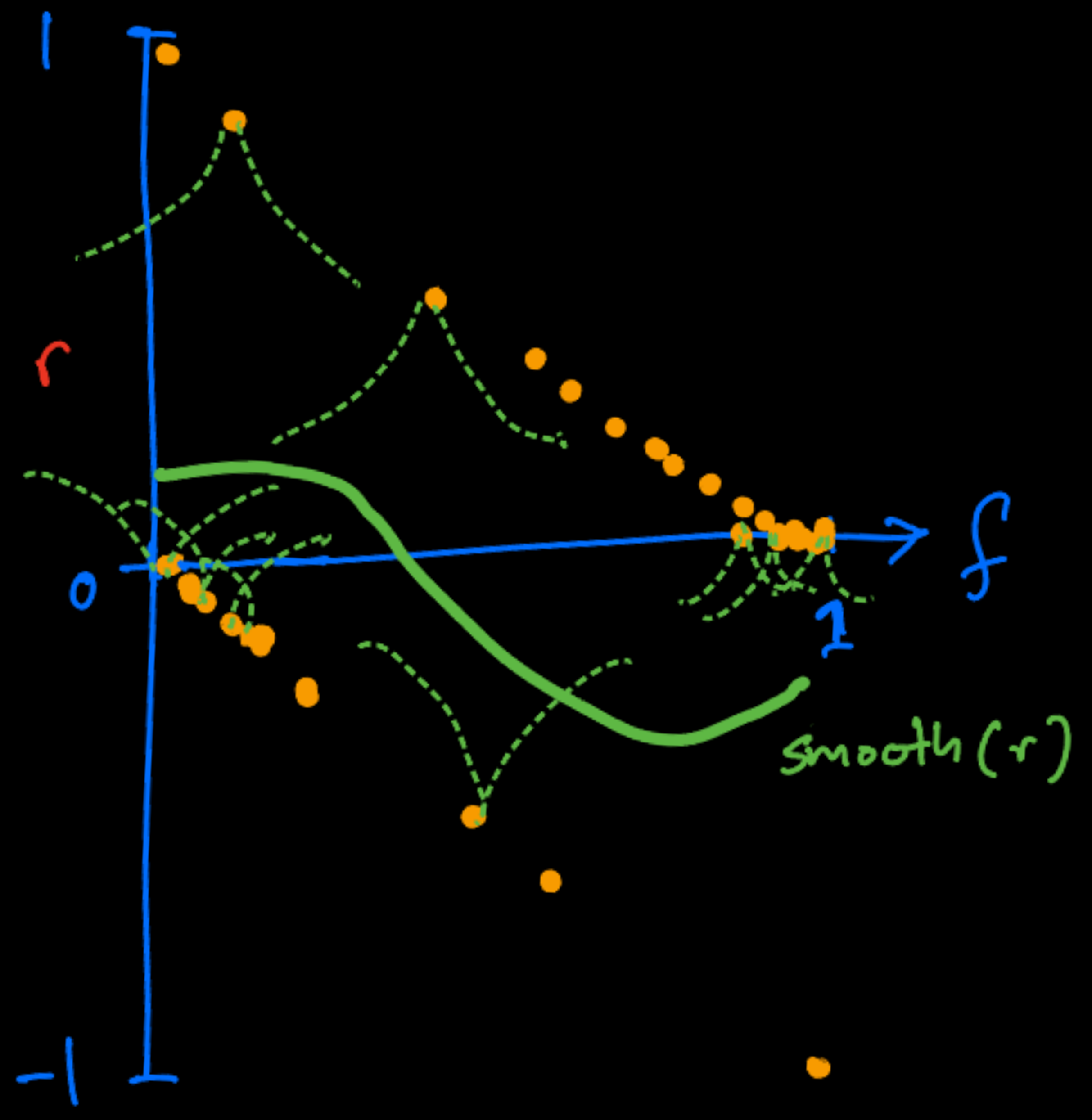
Given: Samples $(f(x_i), y_i) =: (f_i, y_i)$

Residuals: (f_i, r_i) for $r_i := (y_i - f_i)$

$$\widehat{\mathsf{kCE}_{\mathcal{D}}}(f) = \sqrt{\frac{1}{n^2}\sum_{i,j} r_i r_j K(f_i, f_j)} = \|r\|_{K(f,f)}$$

$$= \sqrt{\langle r,\ \mathsf{smooth}_K(r)\rangle}$$





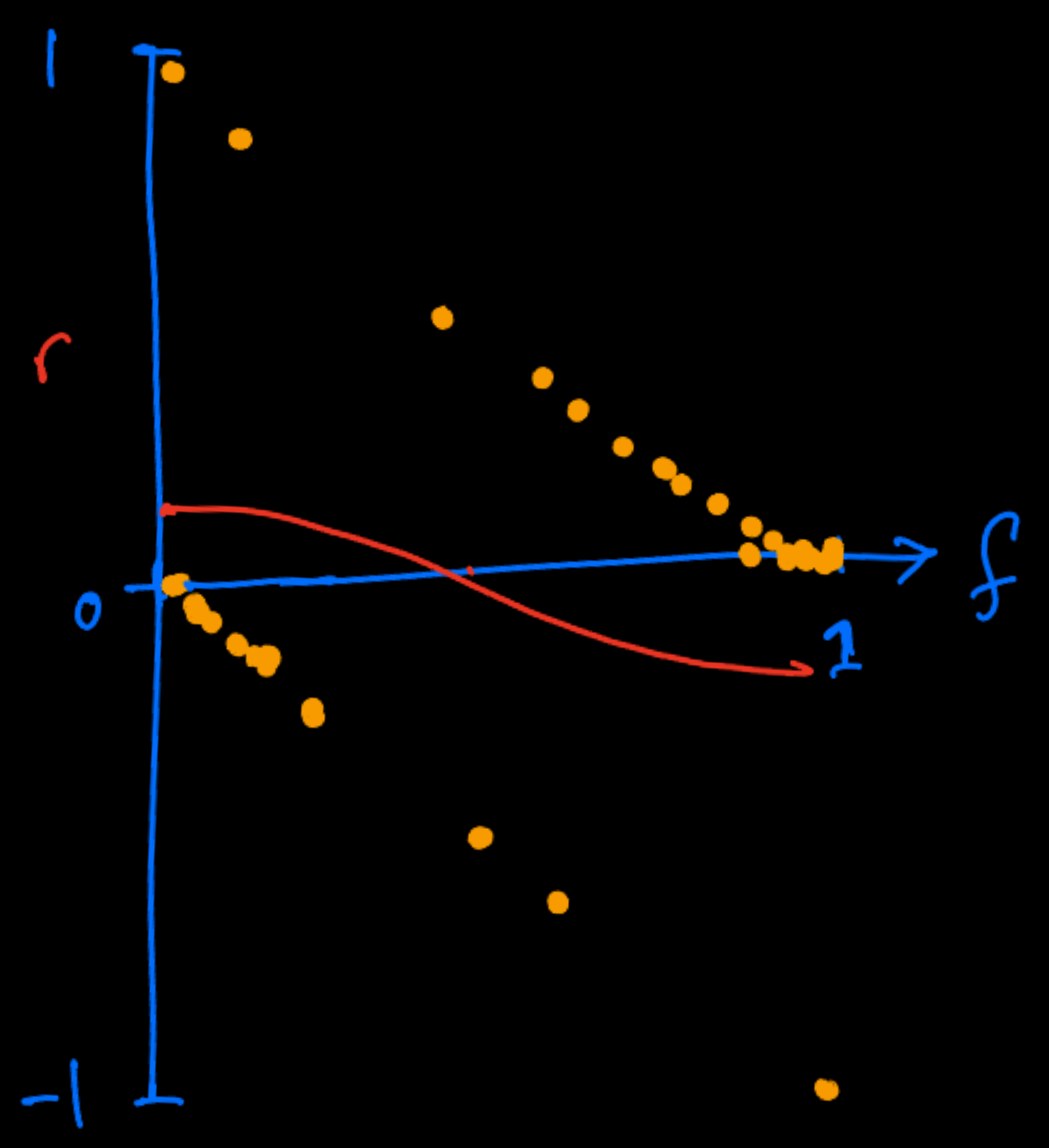
Kernel Calibration Error: Sample Estimation

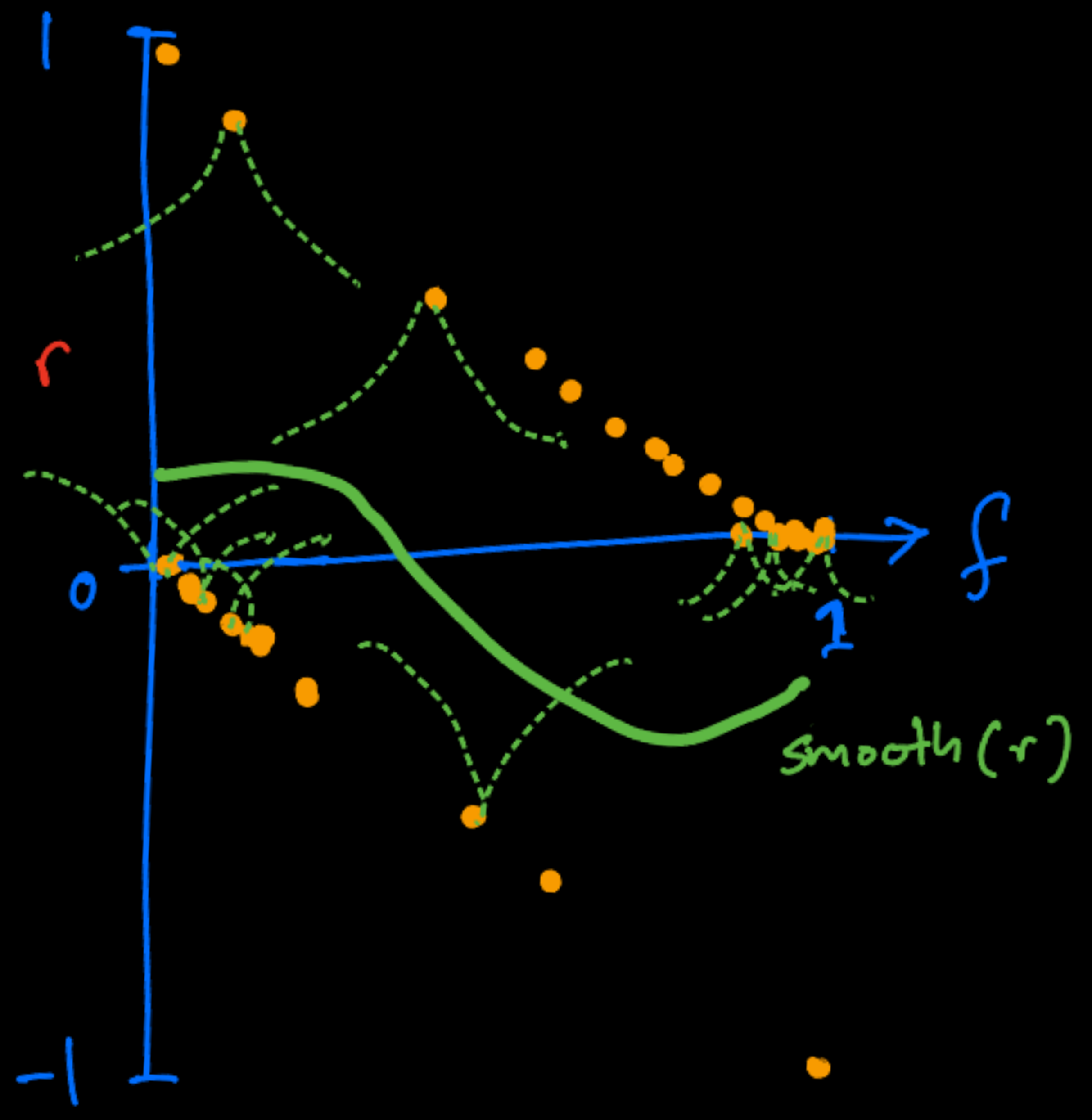
<u>Given:</u> Samples $(f(x_i), y_i) =: (f_i, y_i)$

Residuals: (f_i, r_i) for $r_i := (y_i - f_i)$

$$\widehat{\mathsf{kCE}_{\mathcal{D}}}(f) = \sqrt{\frac{1}{n^2}\sum_{i,j} r_i r_j K(f_i, f_j)} = \|r\|_{K(f,f)}$$

$$= \sqrt{\langle r,\ \mathsf{smooth}_K(r)\rangle}$$



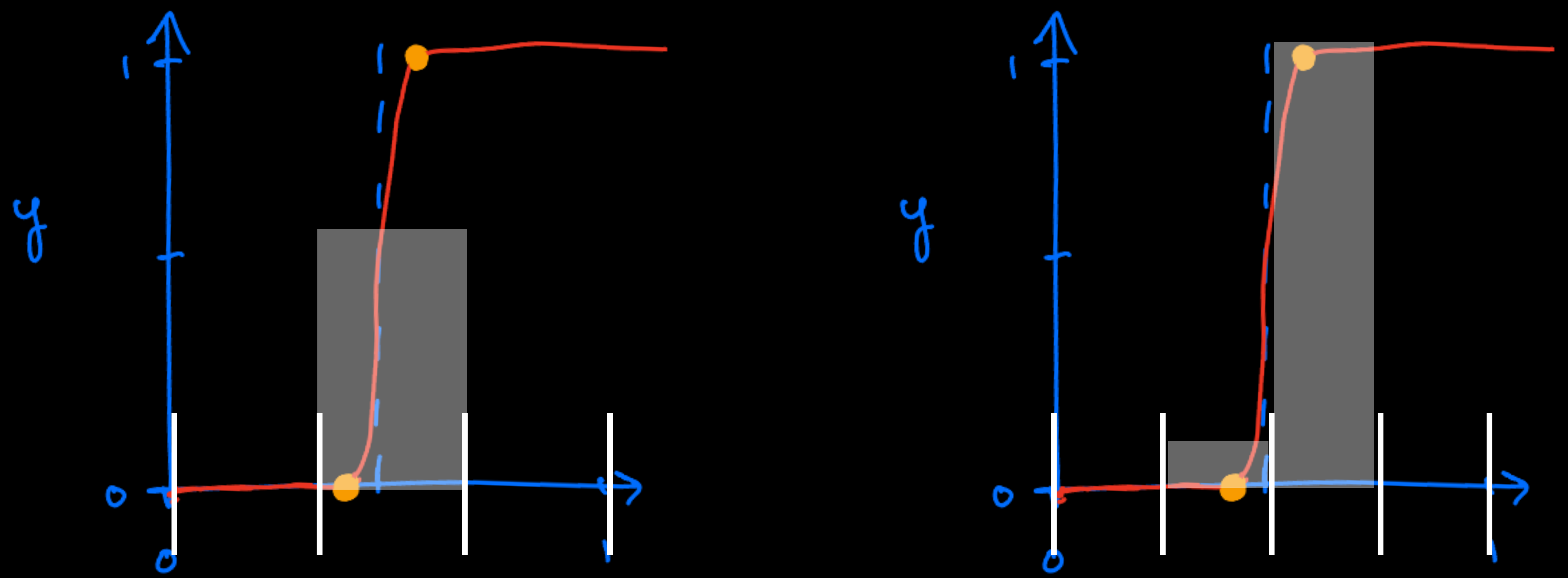


- Linear-time estimation: sub-sample $\Theta(n)$ terms
- Requires **Laplace** kernel

Interval Calibration

$$\mathrm{binnedECE}(f, \mathcal{I}) := \mathrm{ECE}(\mathrm{round}_{\mathcal{I}}(f))$$

binnedECE: Unclear how to choose bins (any fixed choice violates continuity & correctness)



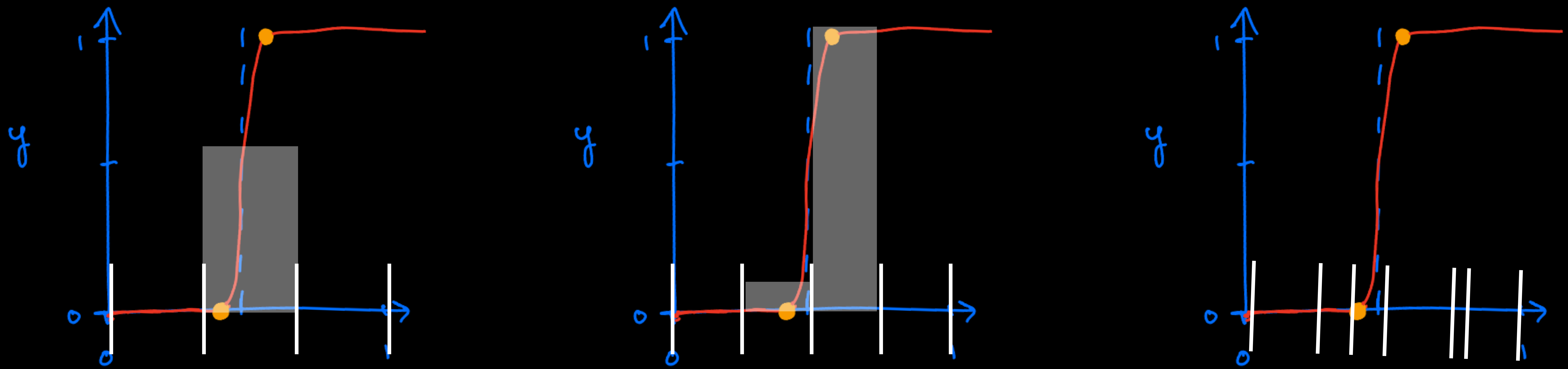
Interval Calibration

$$\mathsf{binnedECE}(f, \mathcal{I}) := \mathsf{ECE}(\mathsf{round}_{\mathcal{I}}(f))$$

binnedECE: Unclear how to choose bins (any fixed choice violates continuity & correctness)

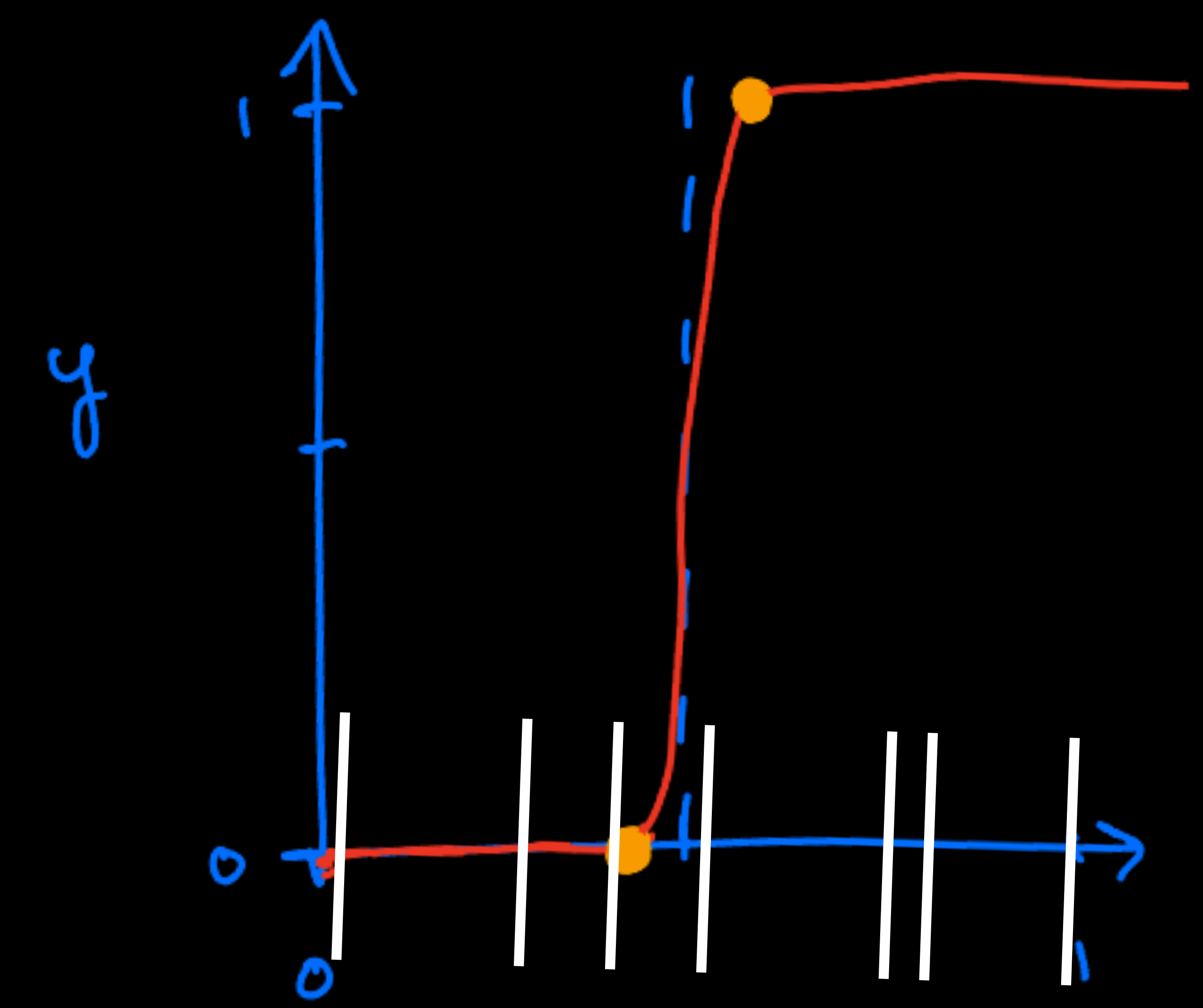
But, adding a "width regularizer" guarantees upper-bound. For all interval-partitions:

$$\mathsf{dCE}(f) \leq \mathsf{binnedECE}(f, \mathcal{I}) + \mathsf{width}(\mathcal{I})$$



Interval Calibration

$$\mathsf{dCE}(f) \leq \mathsf{binnedECE}(f, \mathcal{I}) + \mathsf{width}(\mathcal{I})$$

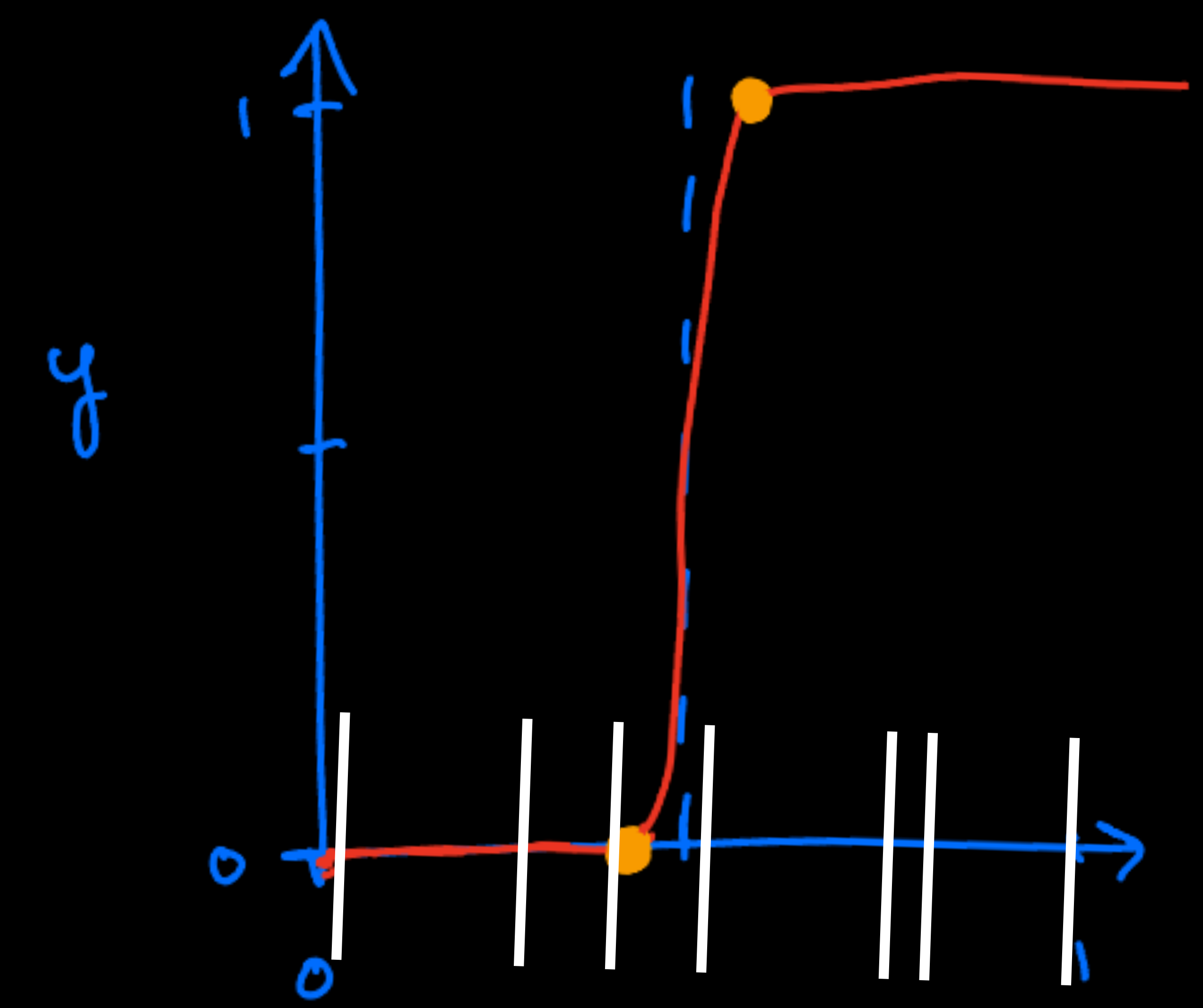


Interval Calibration

$$\mathrm{dCE}(f) \le \mathrm{binnedECE}(f, \mathcal{I}) + \mathrm{width}(\mathcal{I})$$

Best-possible upper-bound:

$$\mathrm{intCE}(f) := \inf_{\mathcal{I}\text{: Interval partition}} \left(\mathrm{binnedECE}(f, \mathcal{I}) + \mathrm{width}(\mathcal{I})\right)$$



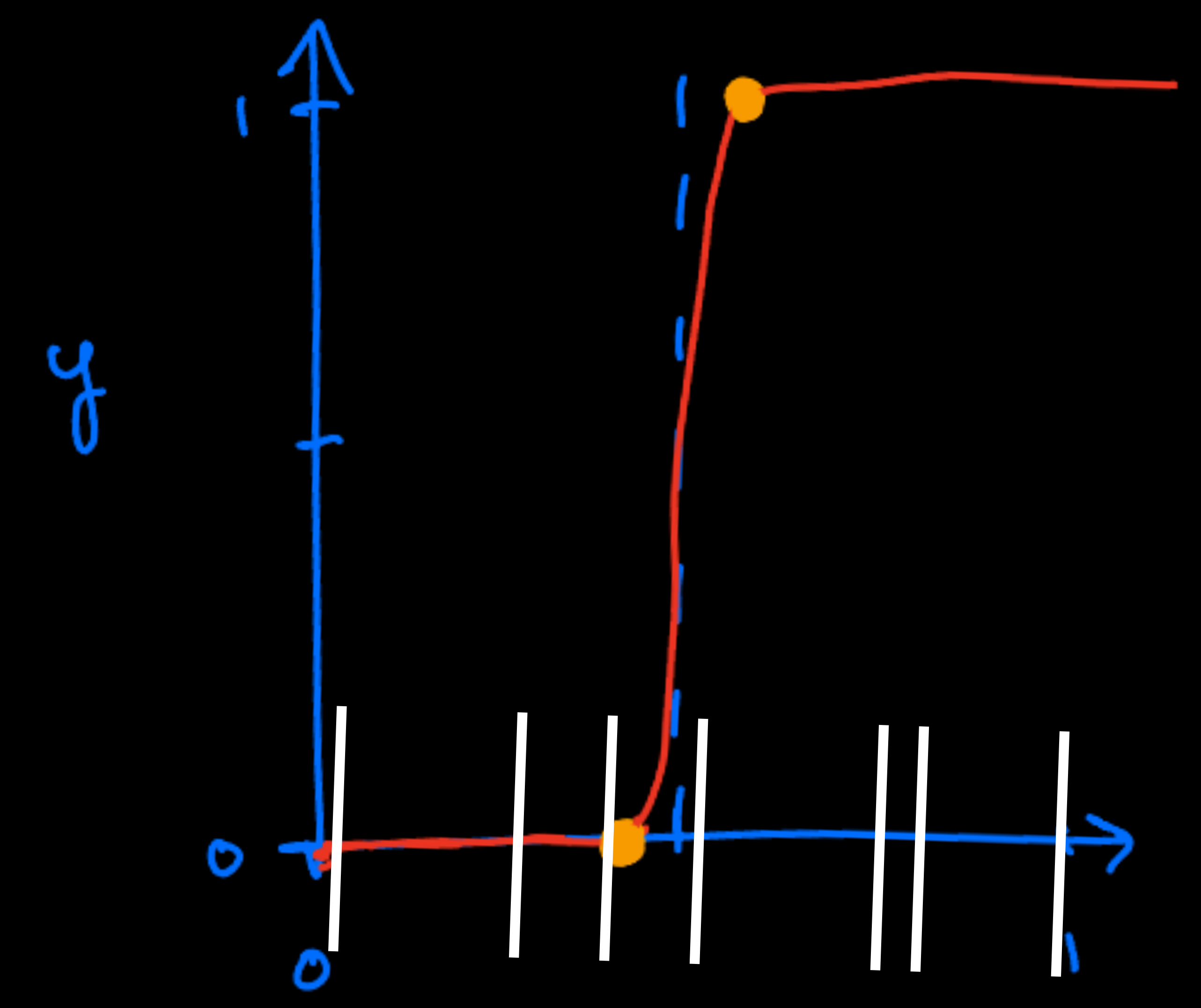
Interval Calibration

$$\mathsf{dCE}(f) \leq \mathsf{binnedECE}(f, \mathcal{I}) + \mathsf{width}(\mathcal{I})$$

Best-possible upper-bound:

$$\mathsf{intCE}(f) := \inf_{\mathcal{I}\text{: Interval partition}} \left(\mathsf{binnedECE}(f, \mathcal{I}) + \mathsf{width}(\mathcal{I})\right)$$

Can we get a lower-bound?



Interval Calibration

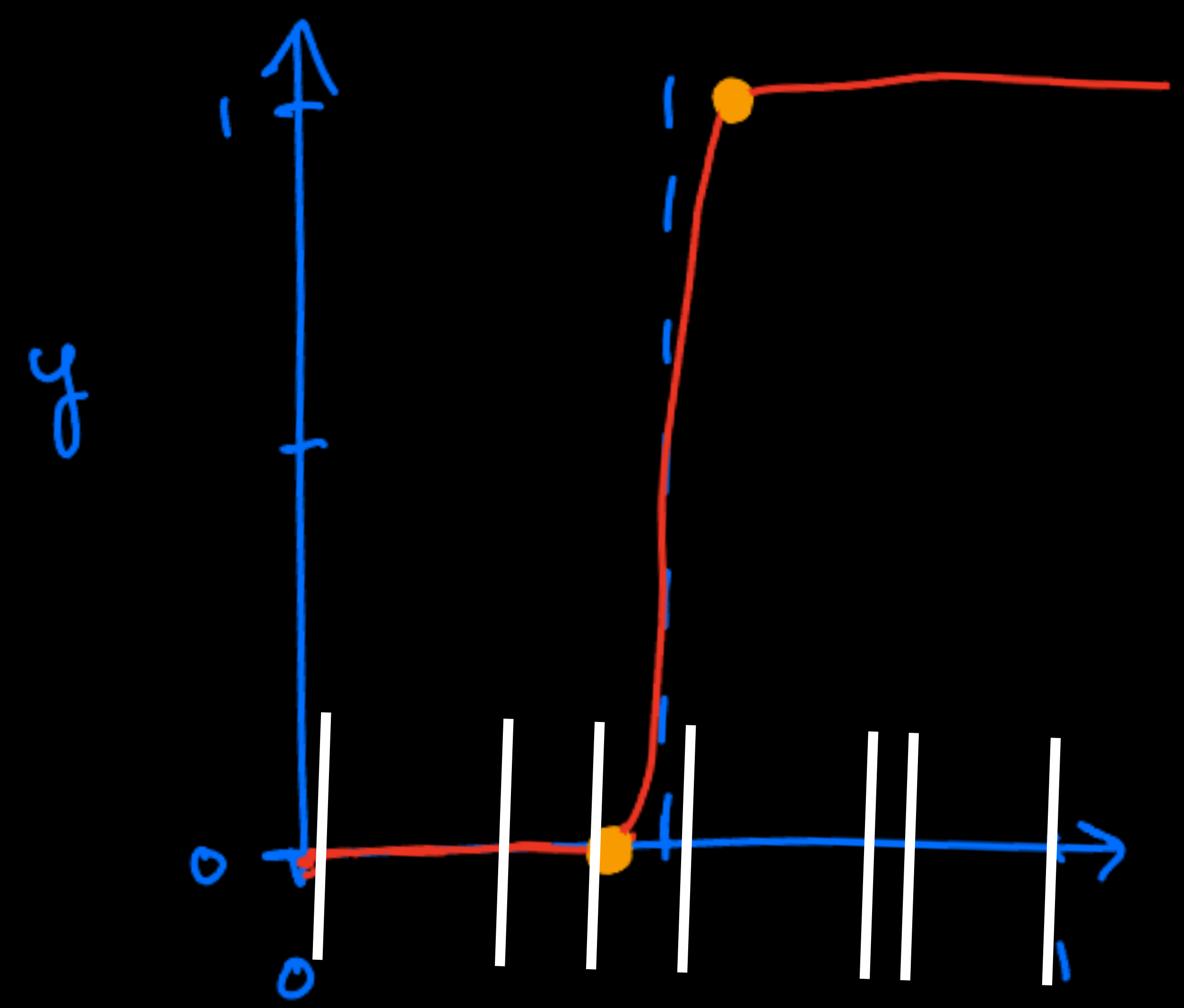
$$\mathrm{dCE}(f) \leq \mathrm{binnedECE}(f, \mathcal{I}) + \mathrm{width}(\mathcal{I})$$

Best-possible upper-bound:

$$\mathrm{intCE}(f) := \inf_{\mathcal{I}\text{: Interval partition}} \left(\mathrm{binnedECE}(f, \mathcal{I}) + \mathrm{width}(\mathcal{I})\right)$$

Can we get a lower-bound?

$$\boxed{\frac{1}{16}\mathrm{intCE}(f)^2 \leq \mathrm{dCE}(f) \leq \mathrm{intCE}(f)}$$



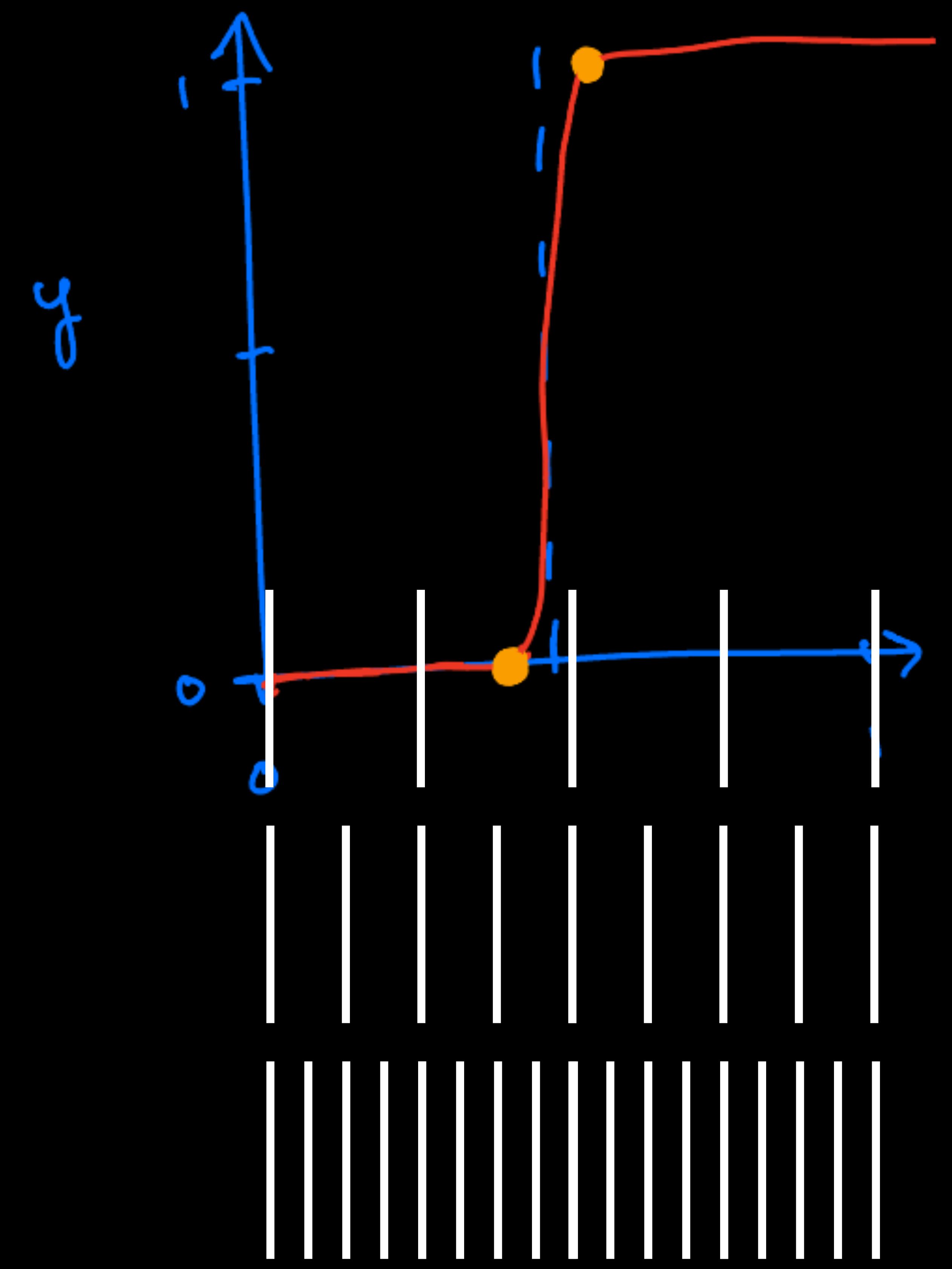
Interval Calibration

$$\mathsf{intCE}(f) := \inf_{\mathcal{I}:\ \text{Interval partition}} \left(\mathsf{binnedECE}(f,\mathcal{I}) + \mathsf{width}(\mathcal{I})\right)$$

Computationally, sufficient to minimize over $i \in \mathbb{N}$:

1. Construct regular intervals of width = 2^{-i}
2. Randomly shift intervals (together)
3. Compute $\mathsf{binnedECE}(f,\mathcal{I}) + \mathsf{width}(\mathcal{I})$

This gives same guarantees!



Practical Takeaways

Measure calibration with either:

1. Kernel Calibration Error

2. Interval Calibration Error

or, if you must use binnedECE, add max-interval-width “regularizer”

In Practice: kCE ≈ binnedECE