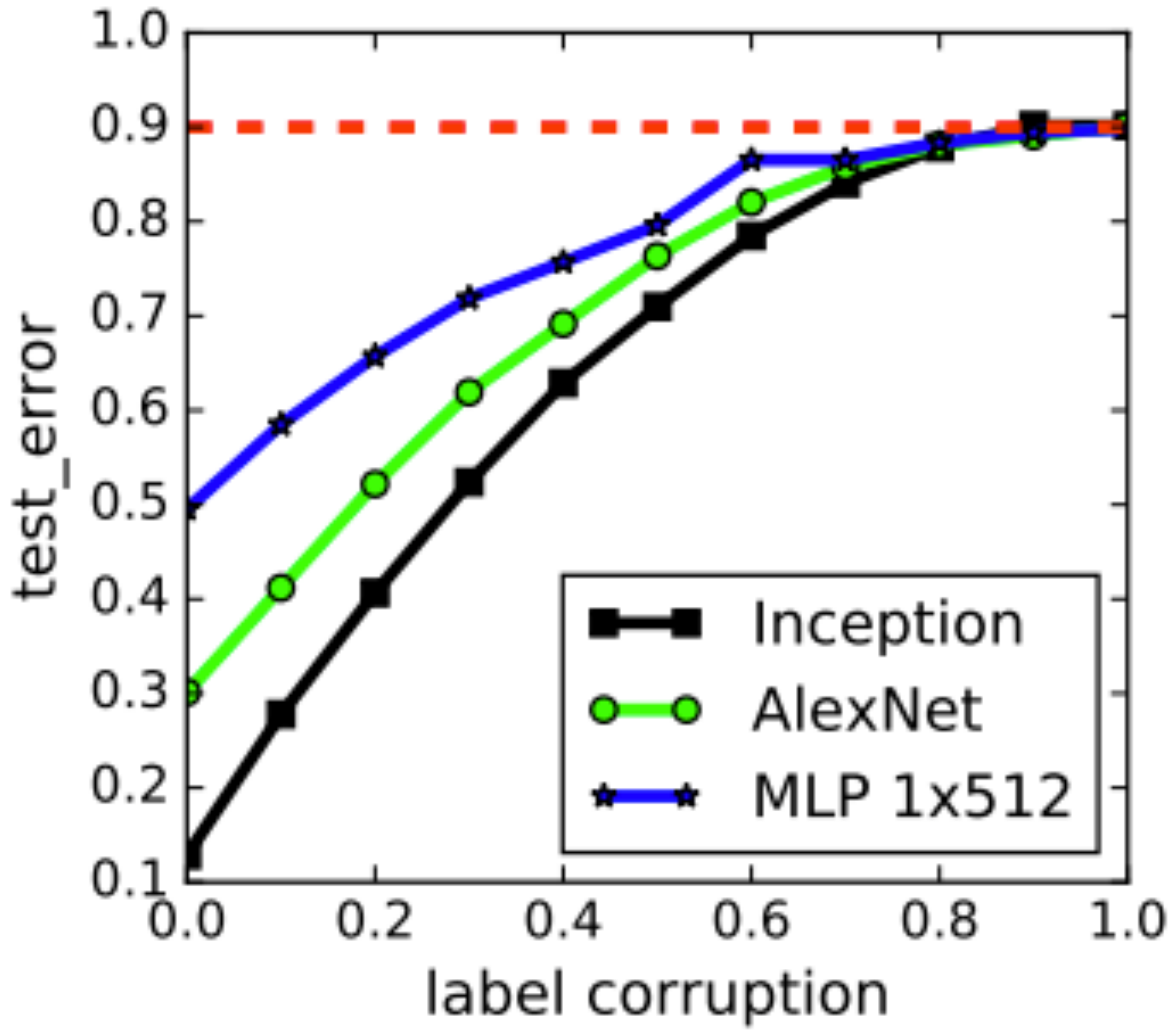
The Devil is in the Tails and other Stories of Interpolation

Niladri Chatterji
Stanford University

with Tatsunori Hashimoto, Saminul Haque, Philip Long and Alexander Wang

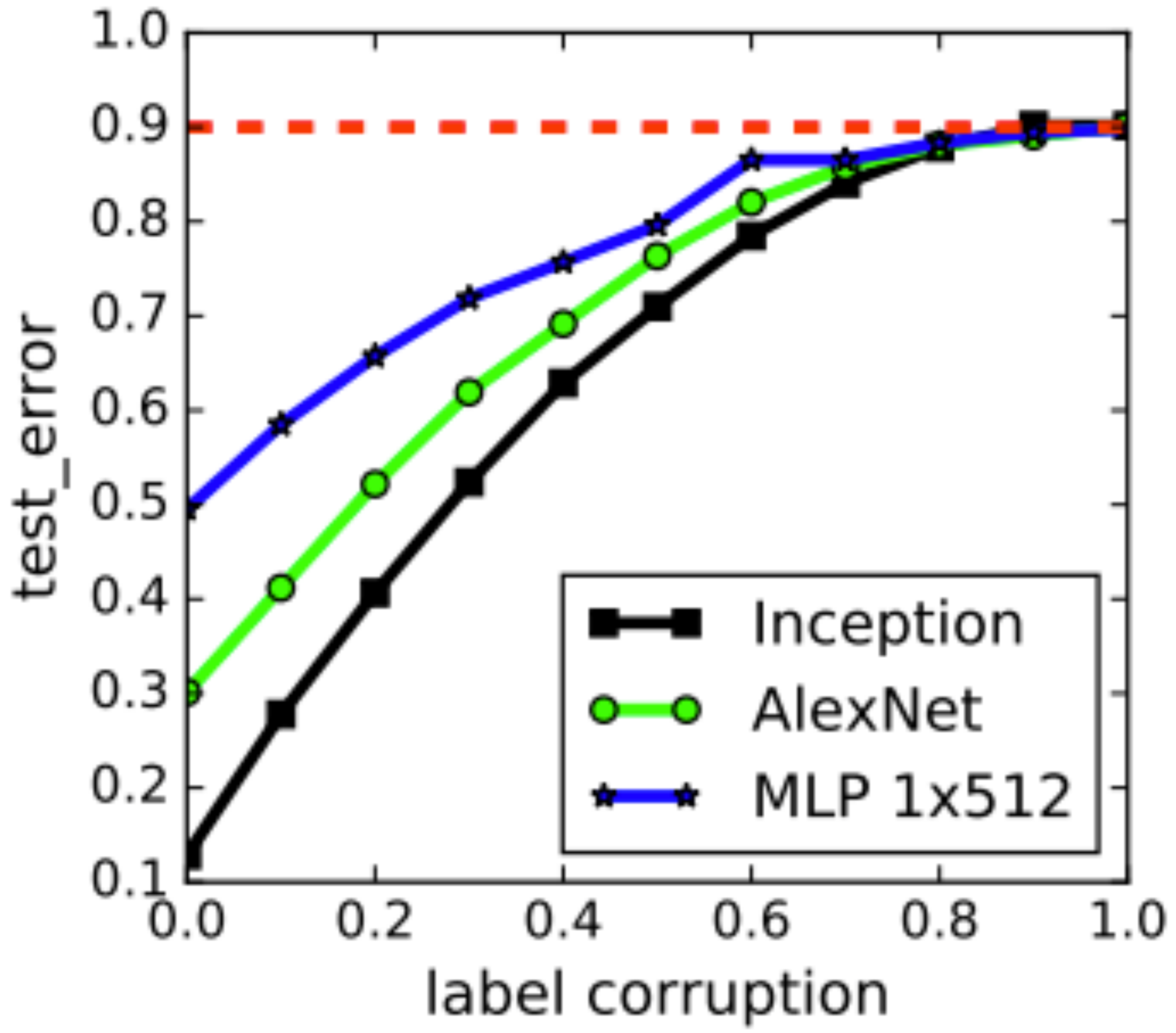
Benign Overfitting in the Presence of Noise



(Zhang et al. 2016)

Deep networks generalize well even when

Benign Overfitting in the Presence of Noise

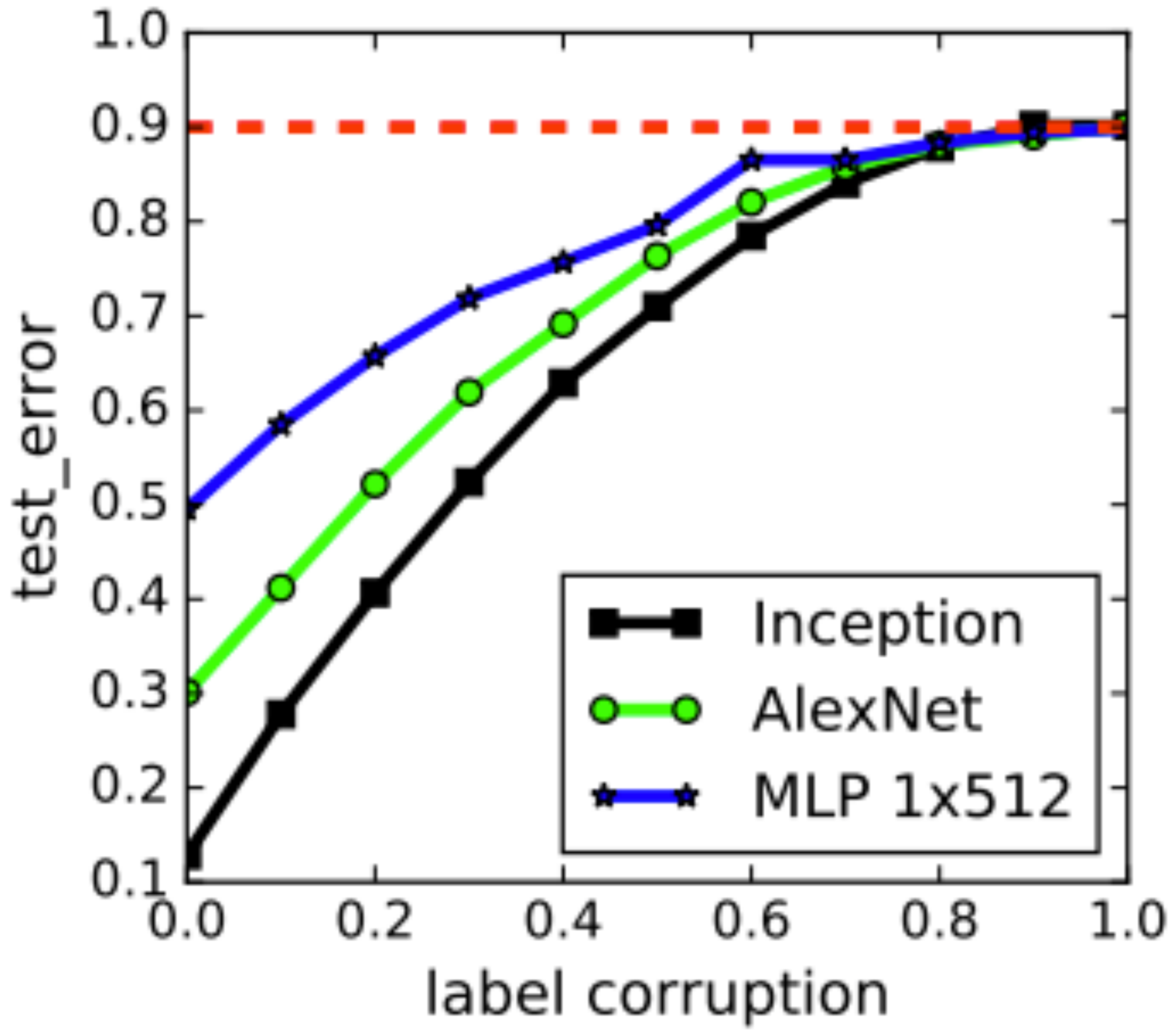


(Zhang et al. 2016)

Deep networks generalize well even when

- data has misclassification noise

Benign Overfitting in the Presence of Noise

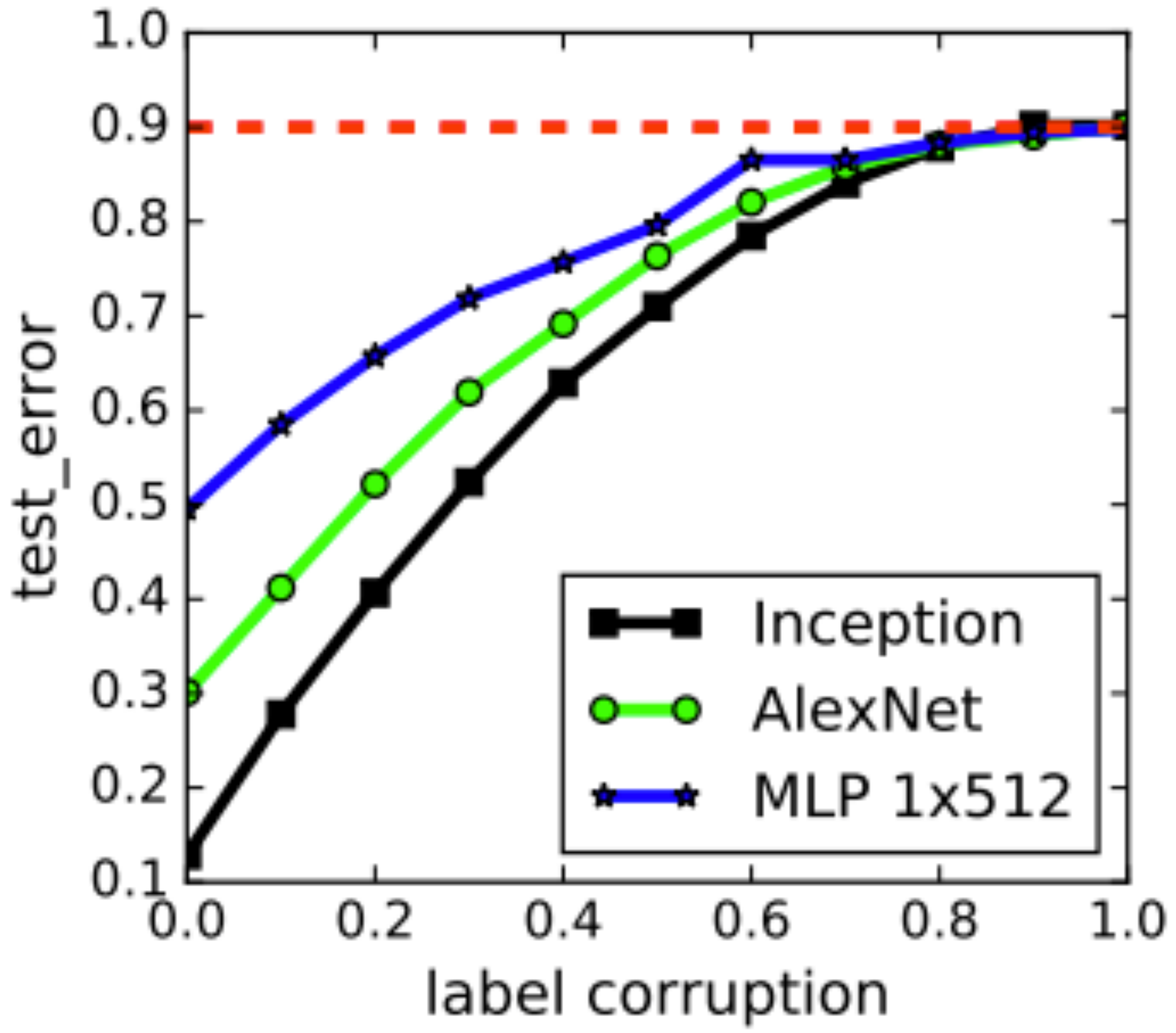


(Zhang et al. 2016)

Deep networks generalize well even when

- data has misclassification noise
- model is overparameterized

Benign Overfitting in the Presence of Noise

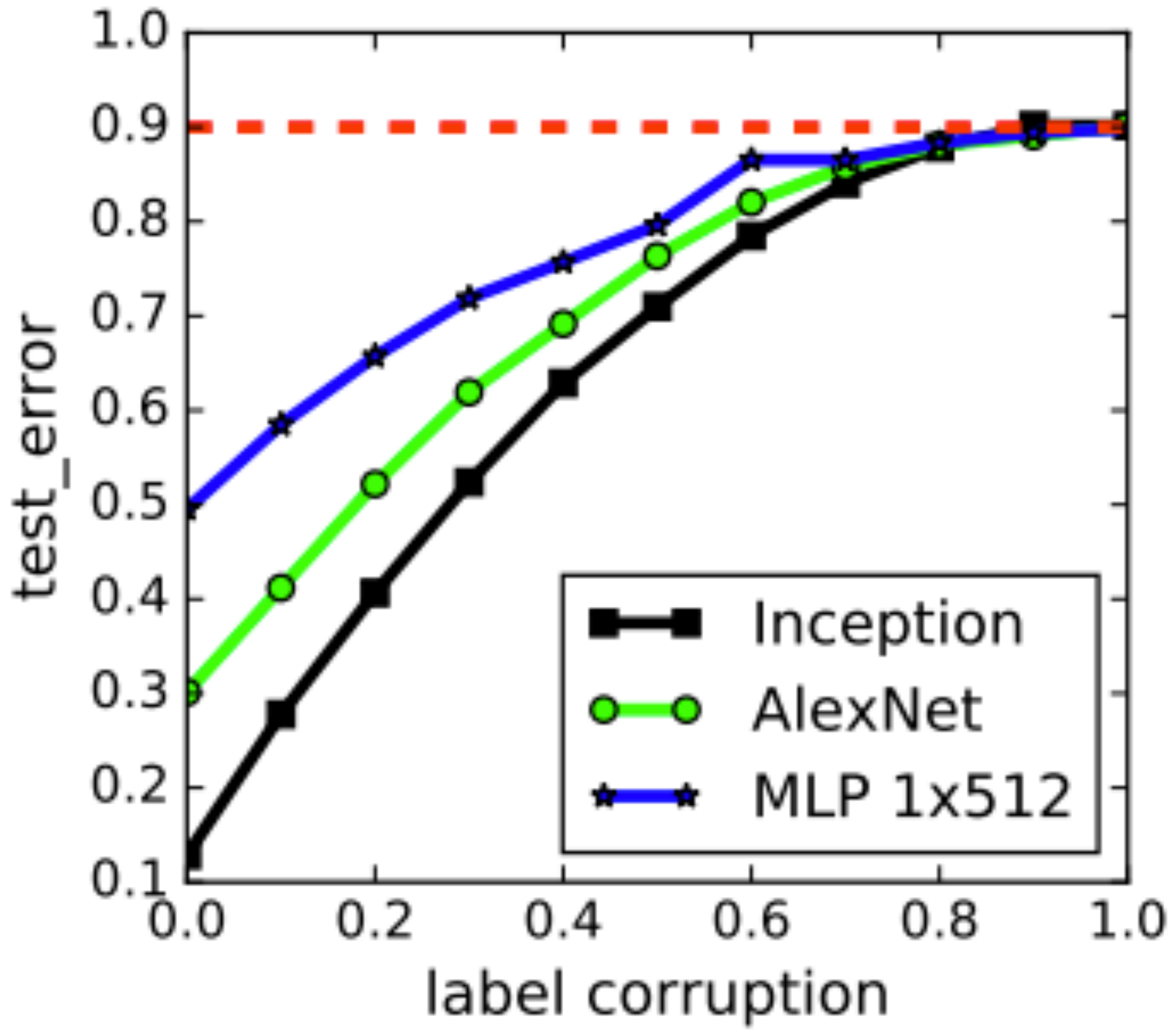


(Zhang et al. 2016)

Deep networks generalize well even when

- data has misclassification noise
- model is overparameterized
- not regularized

Benign Overfitting in the Presence of Noise

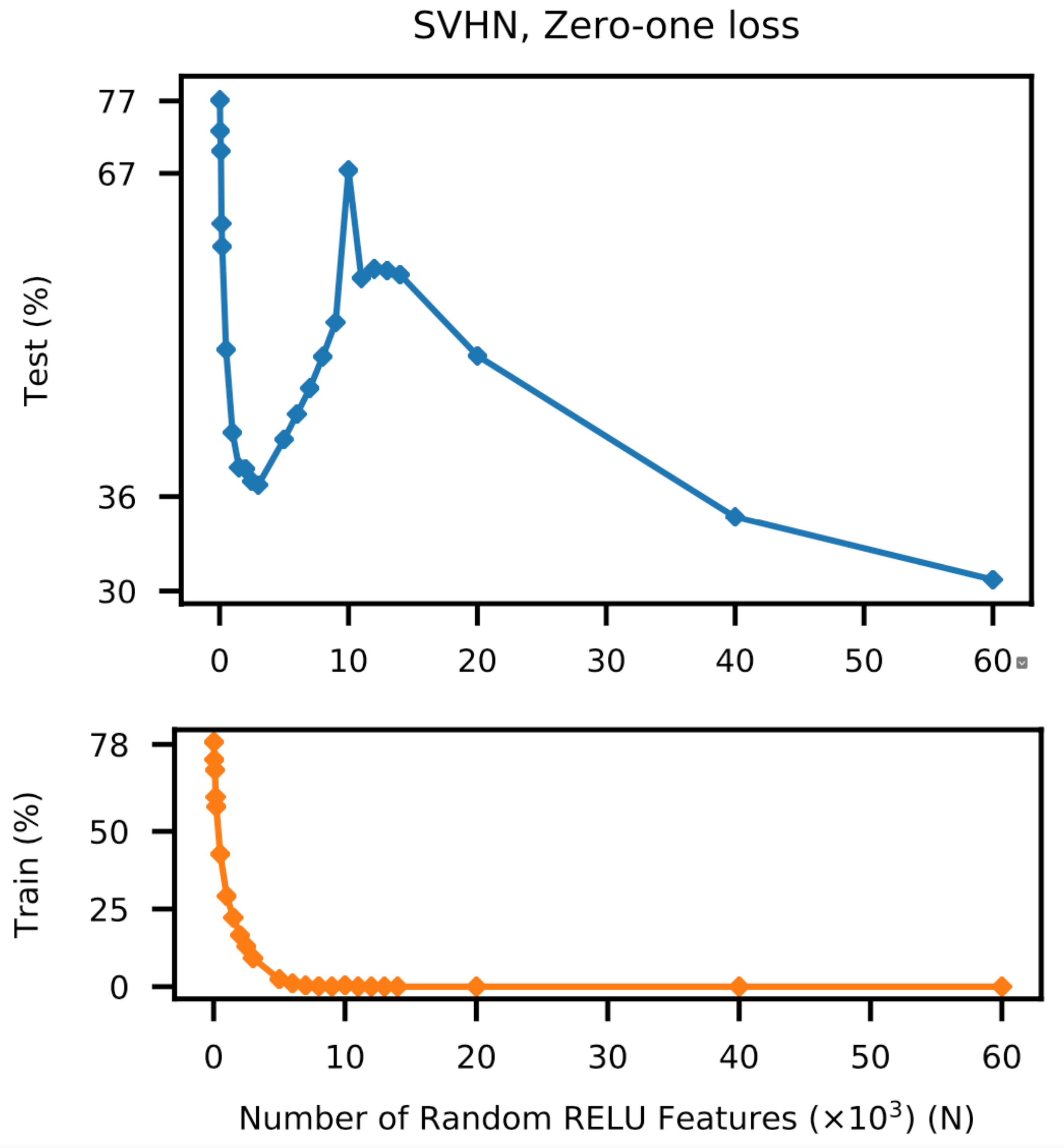


(Zhang et al. 2016)

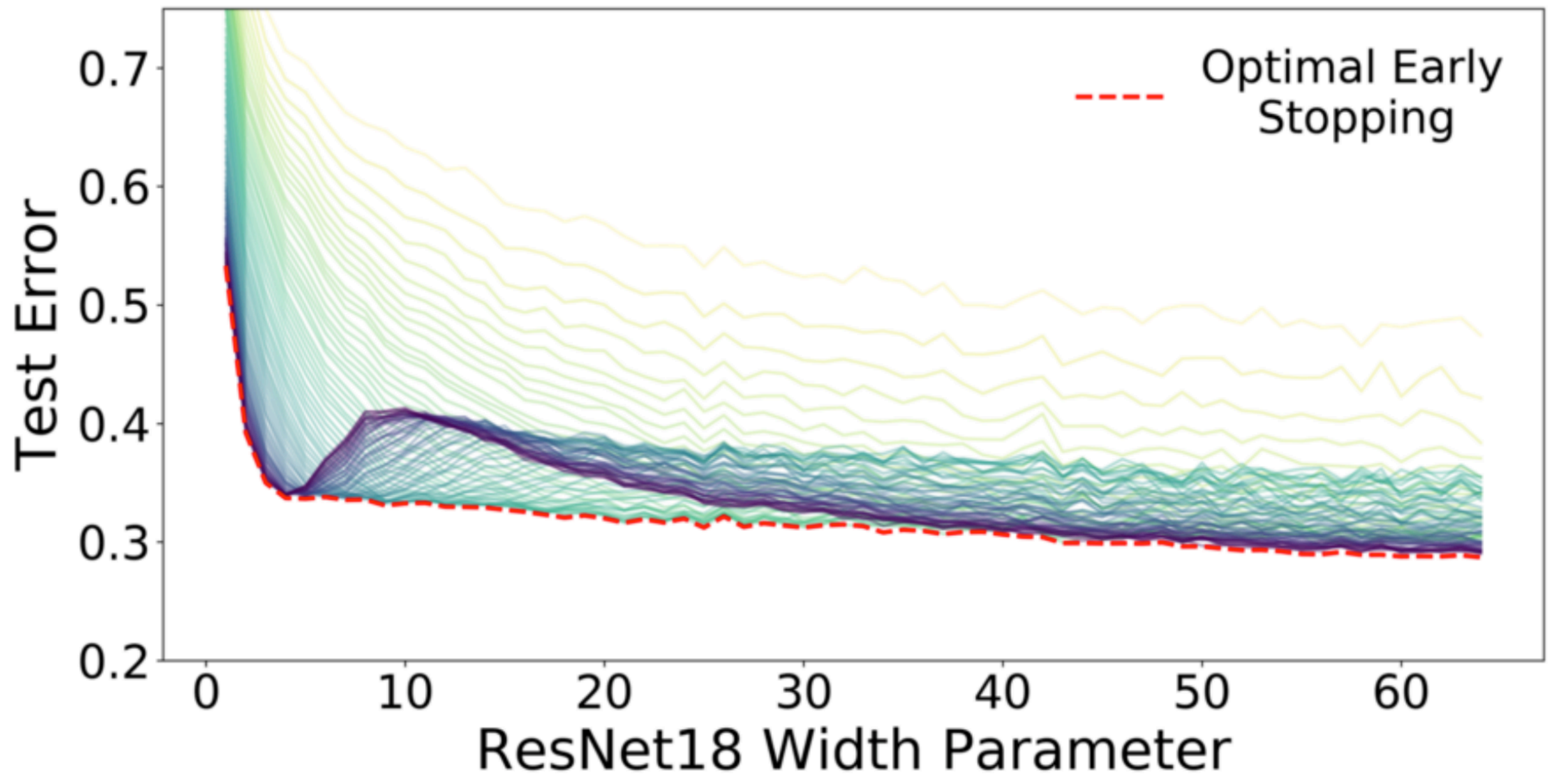
Deep networks generalize well even when

- data has misclassification noise
- model is overparameterized
- not regularized
- trained to zero training loss via SGD

Interpolating Training Data can be beneficial



(Belkin et al. 2018)



(Nakkiran et al. 2021)

Interpolating Training Data can be beneficial

Interpolation is helpful when

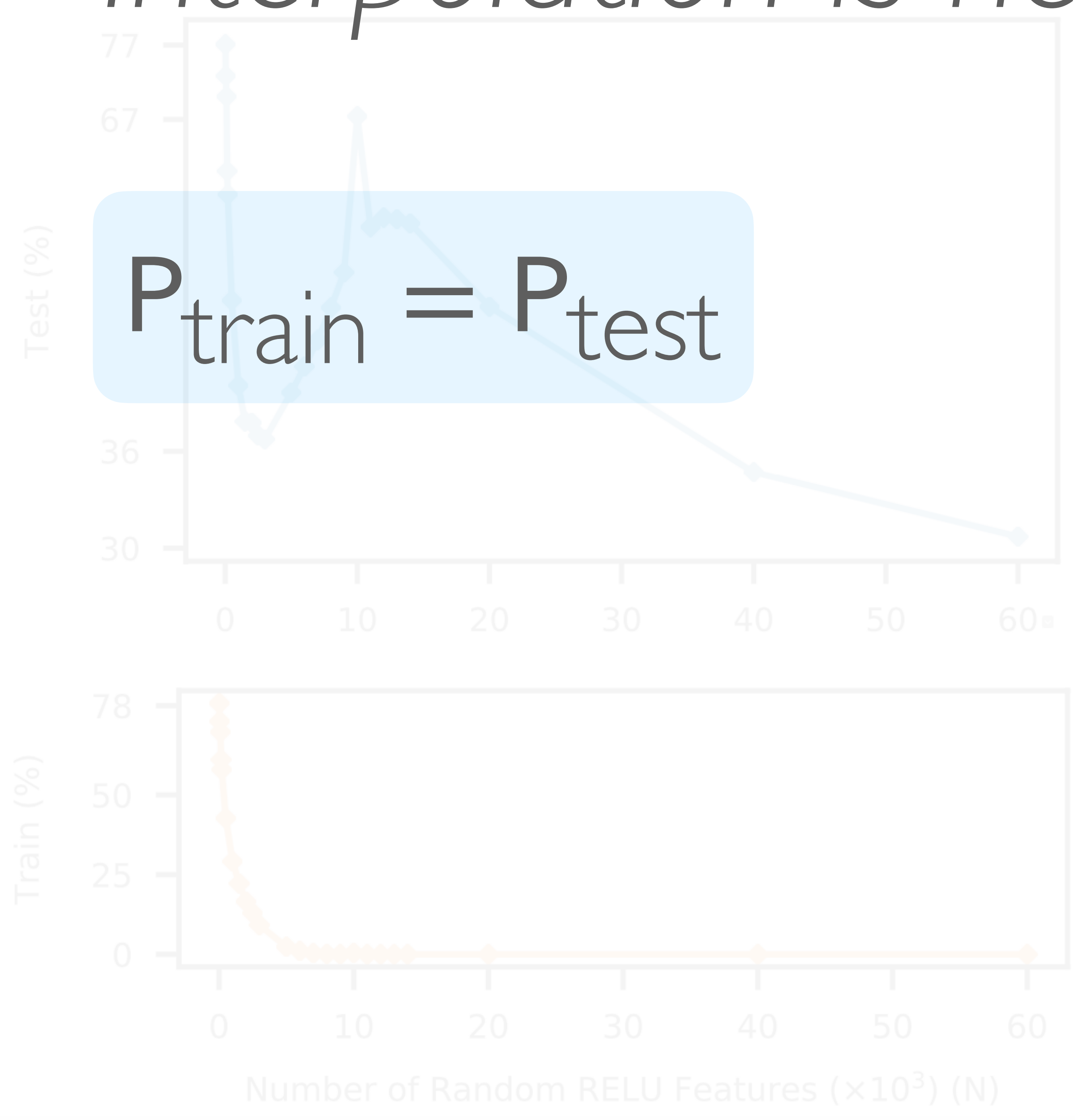
(Belkin et al. 2018)

(Nakkiran et al. 2021)

Interpolating Training Data can be beneficial

Interpolation is helpful when

$P_{train} = P_{test}$



(Belkin et al. 2018)

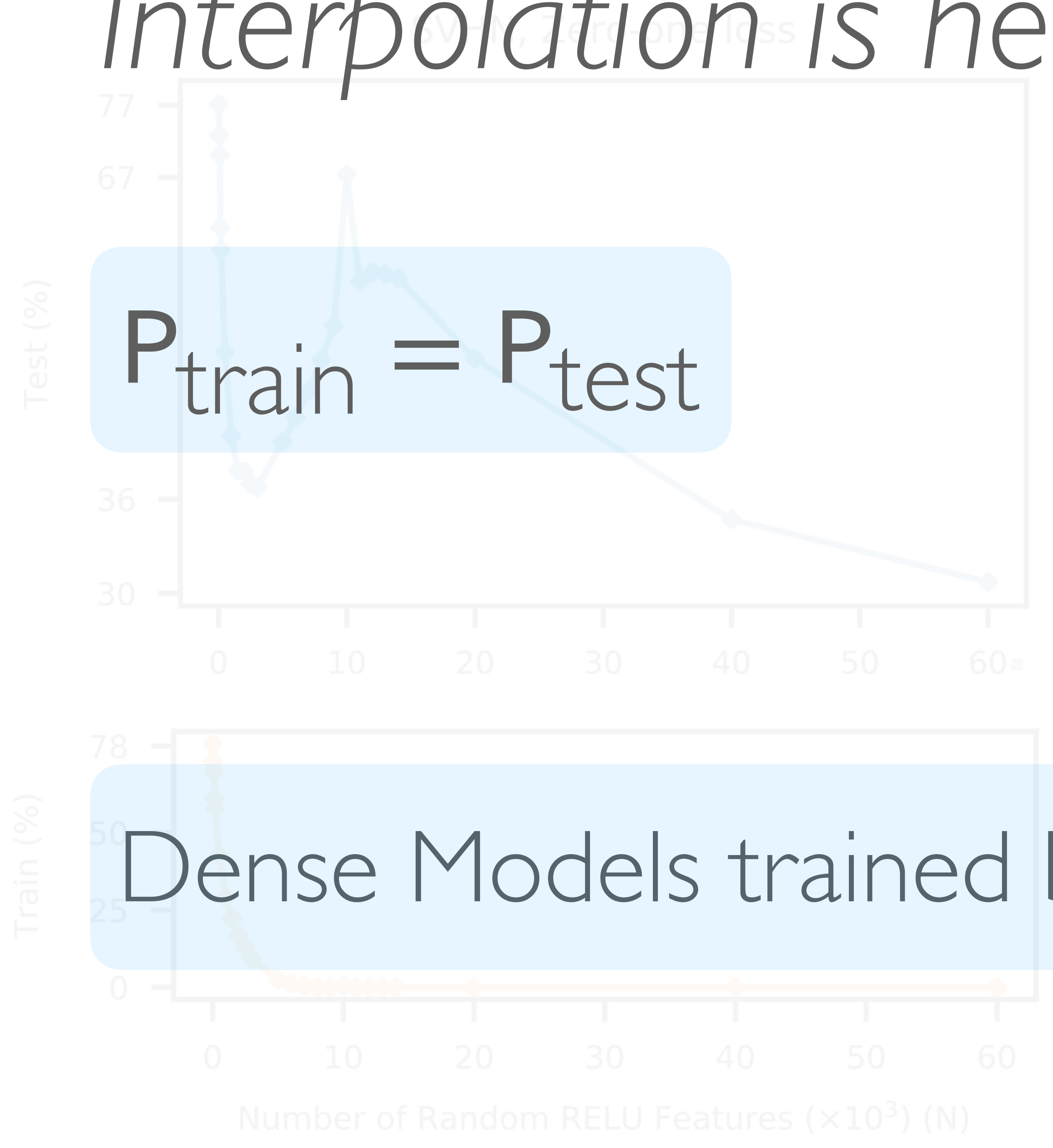
(Nakkiran et al. 2021)

Interpolating Training Data can be beneficial

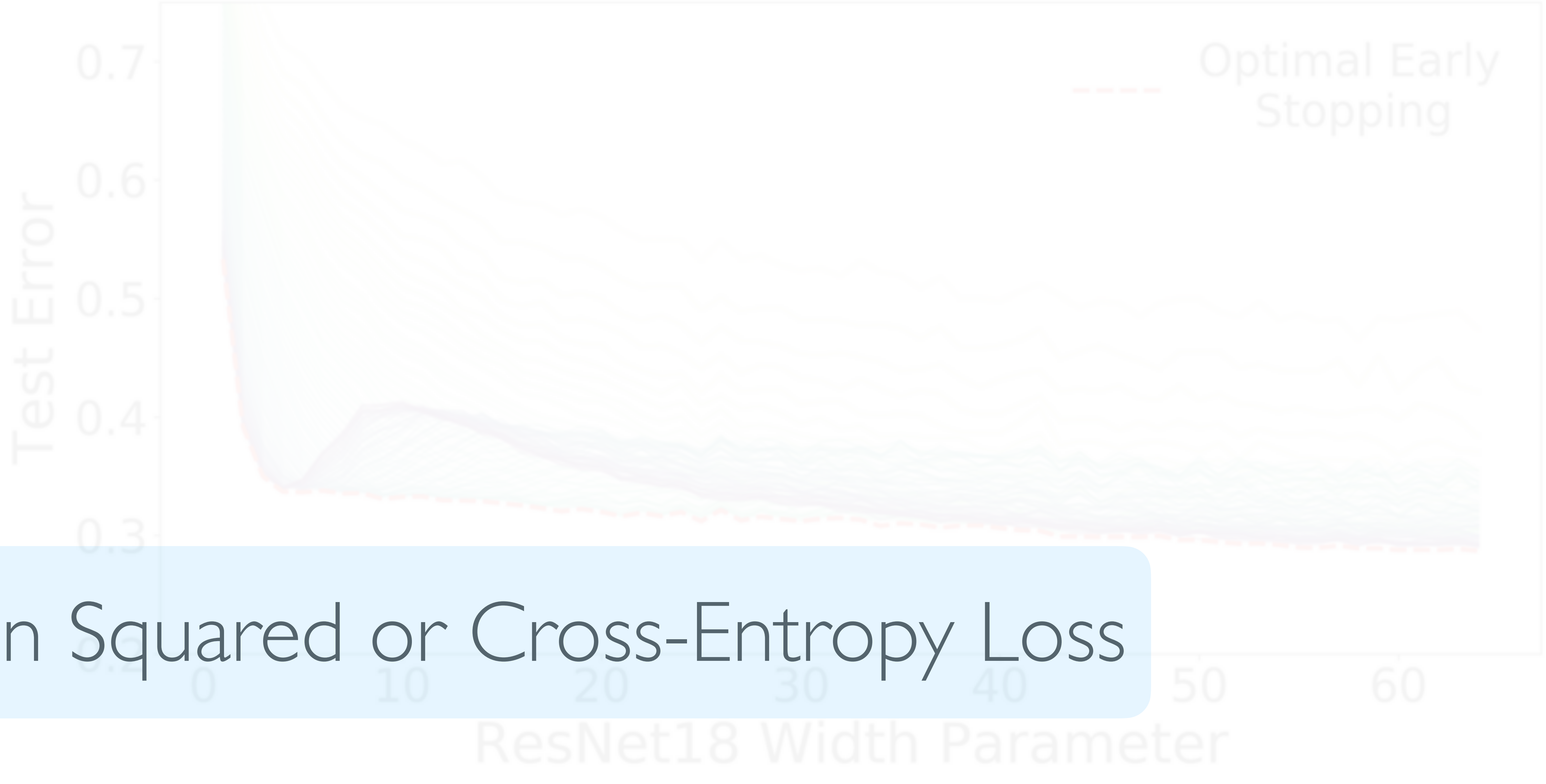
Interpolation is helpful when

$P_{train} = P_{test}$

Dense Models trained by SGD on Squared or Cross-Entropy Loss



(Belkin et al. 2018)



(Nakkiran et al. 2021)

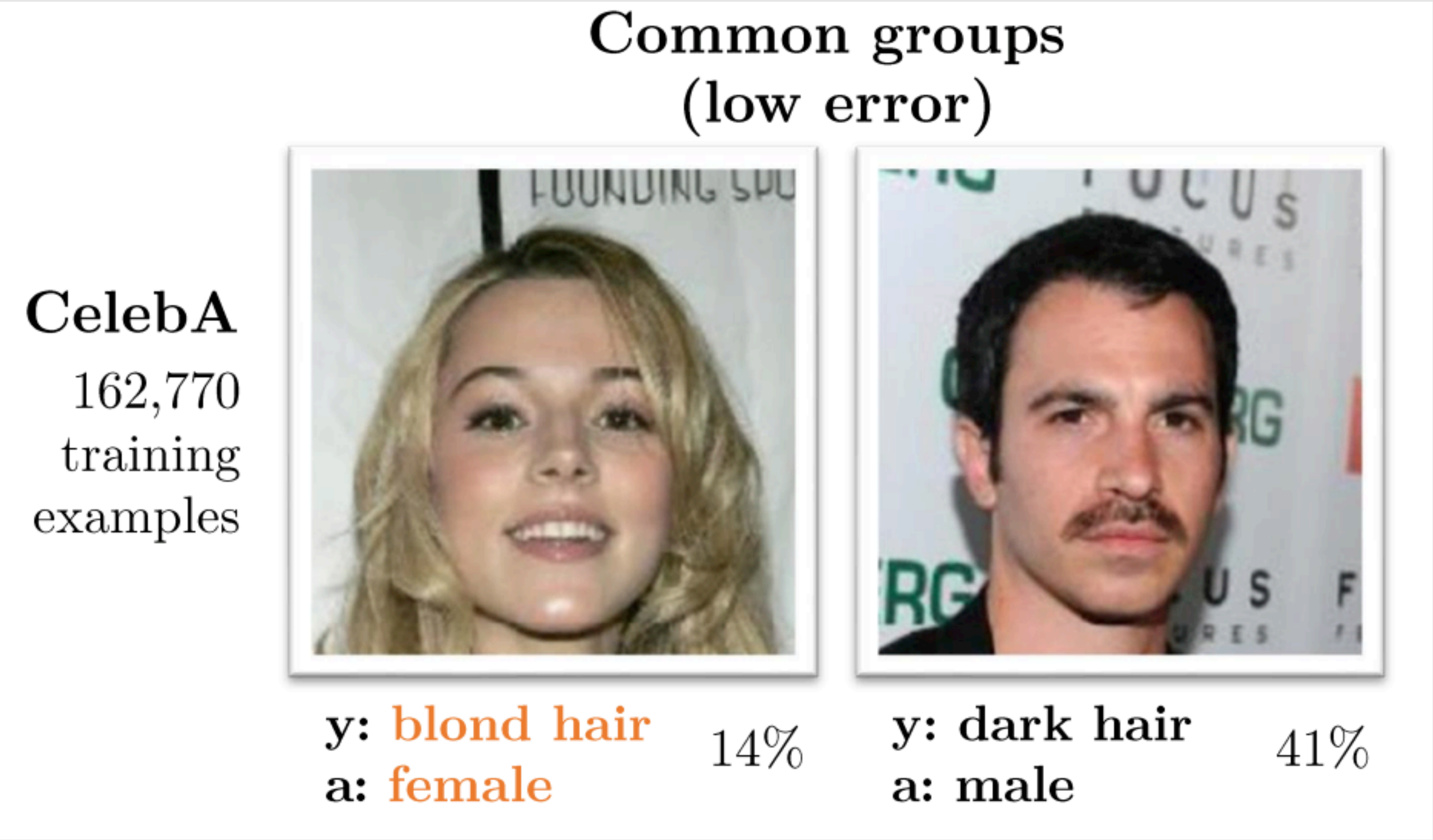
What happens when veer off this standard path?

What happens when veer off this standard path?

Vignette 1: Interpolating Classifiers under shift $P_{train} \neq P_{test}$

What happens when veer off this standard path?

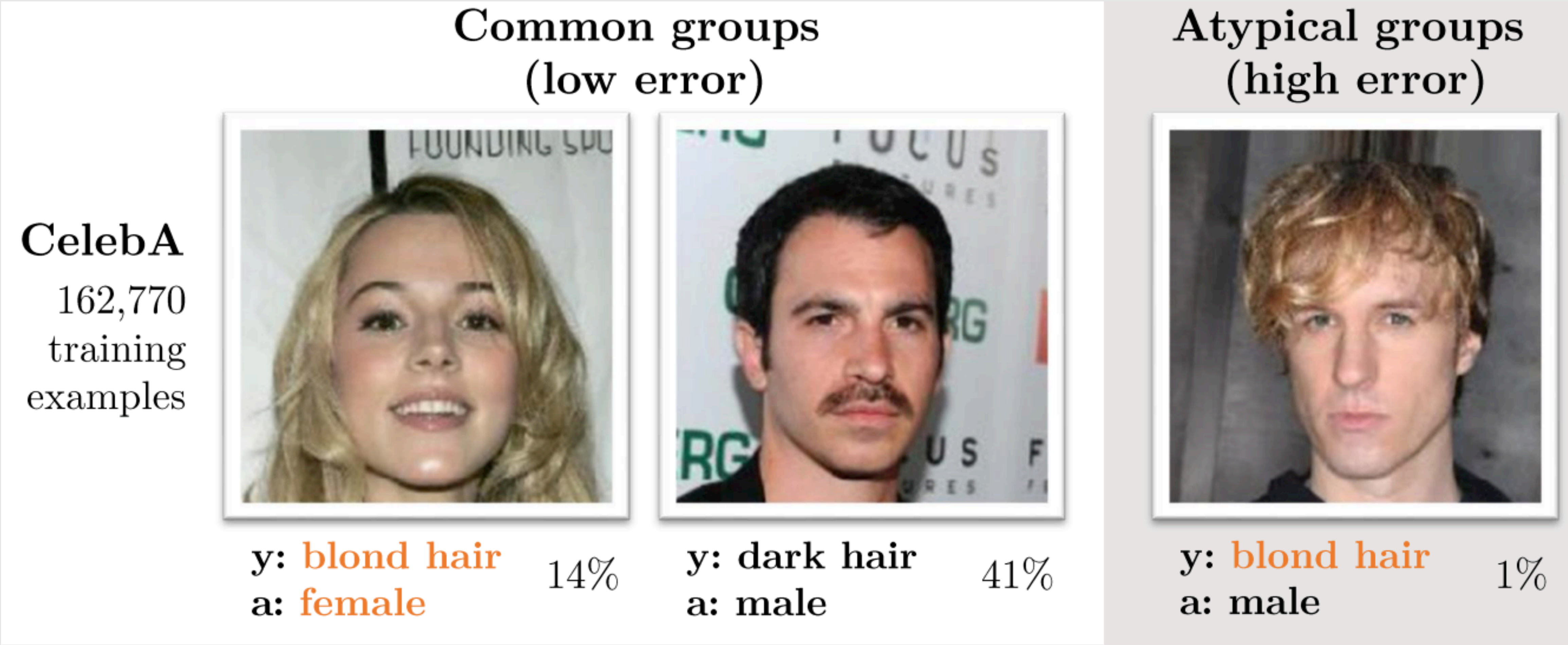
Vignette 1: Interpolating Classifiers under shift $P_{train} \neq P_{test}$



(Sagawa et al. 2020)

What happens when veer off this standard path?

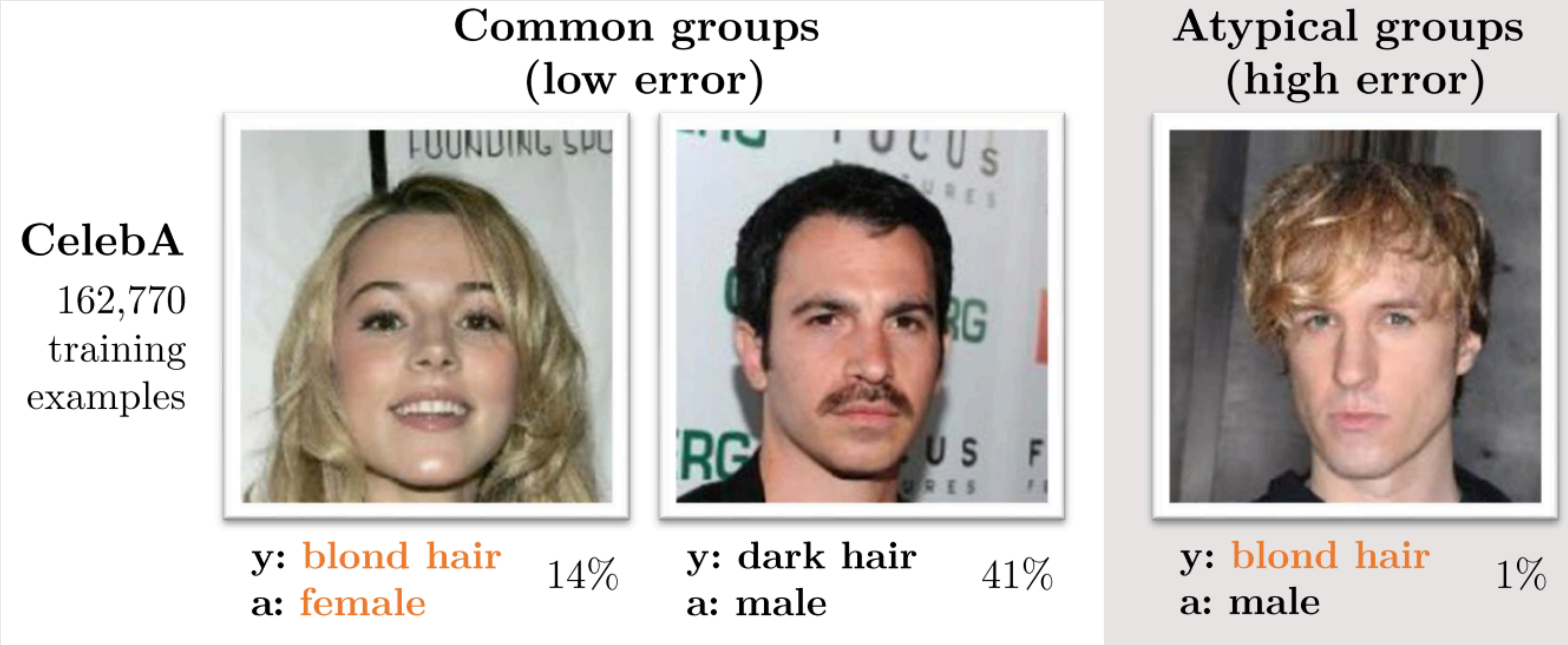
Vignette 1: Interpolating Classifiers under shift $P_{train} \neq P_{test}$



(Sagawa et al. 2020)

What happens when veer off this standard path?

Vignette 1: Interpolating Classifiers under shift $P_{train} \neq P_{test}$

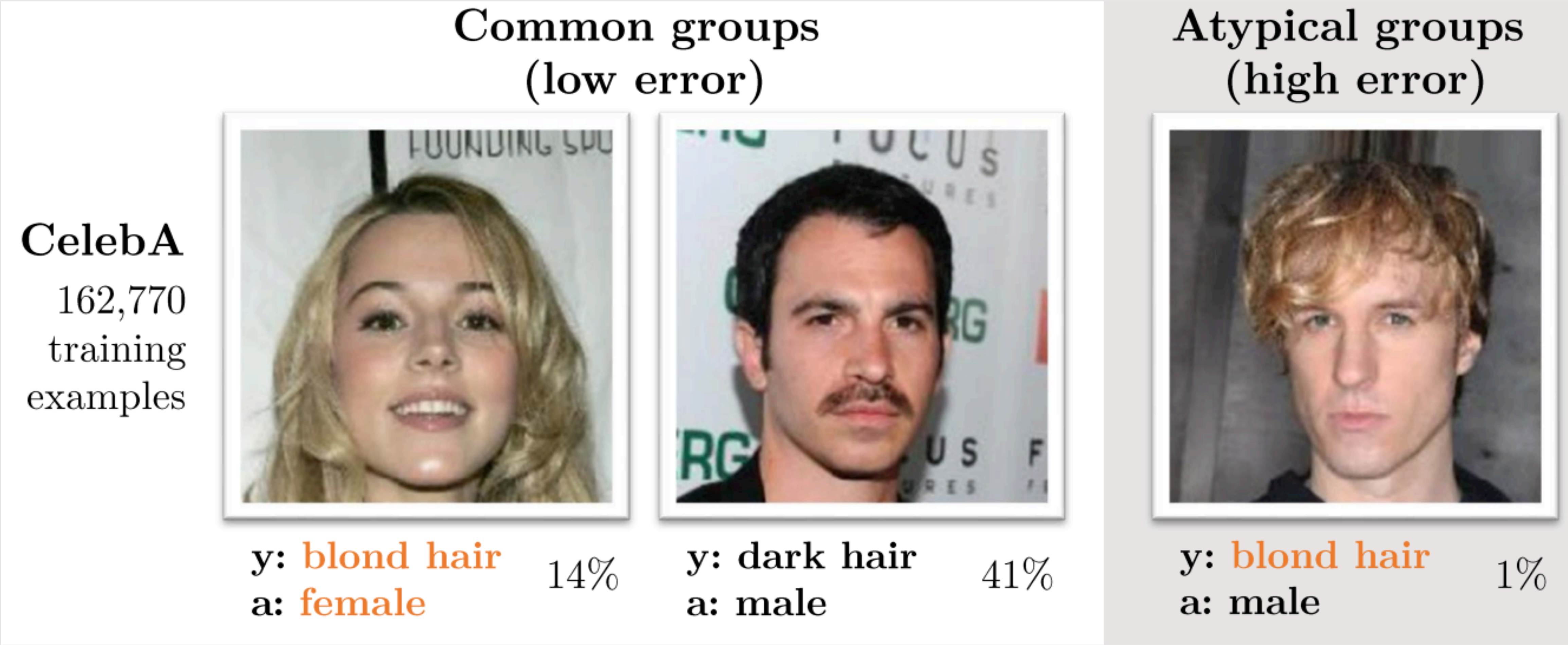


(Sagawa et al. 2020)

- P_{train} is an imbalanced mixture of the groups

What happens when veer off this standard path?

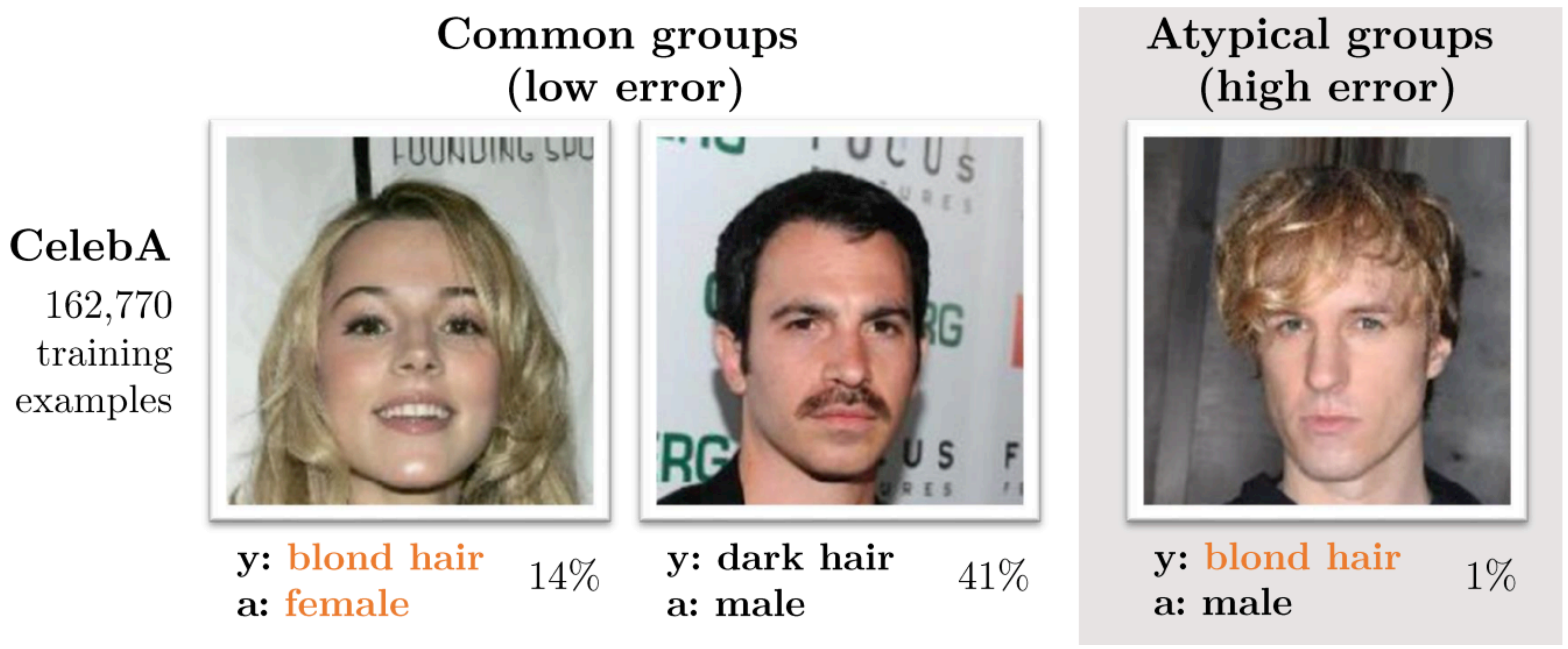
Vignette 1: Interpolating Classifiers under shift $P_{train} \neq P_{test}$



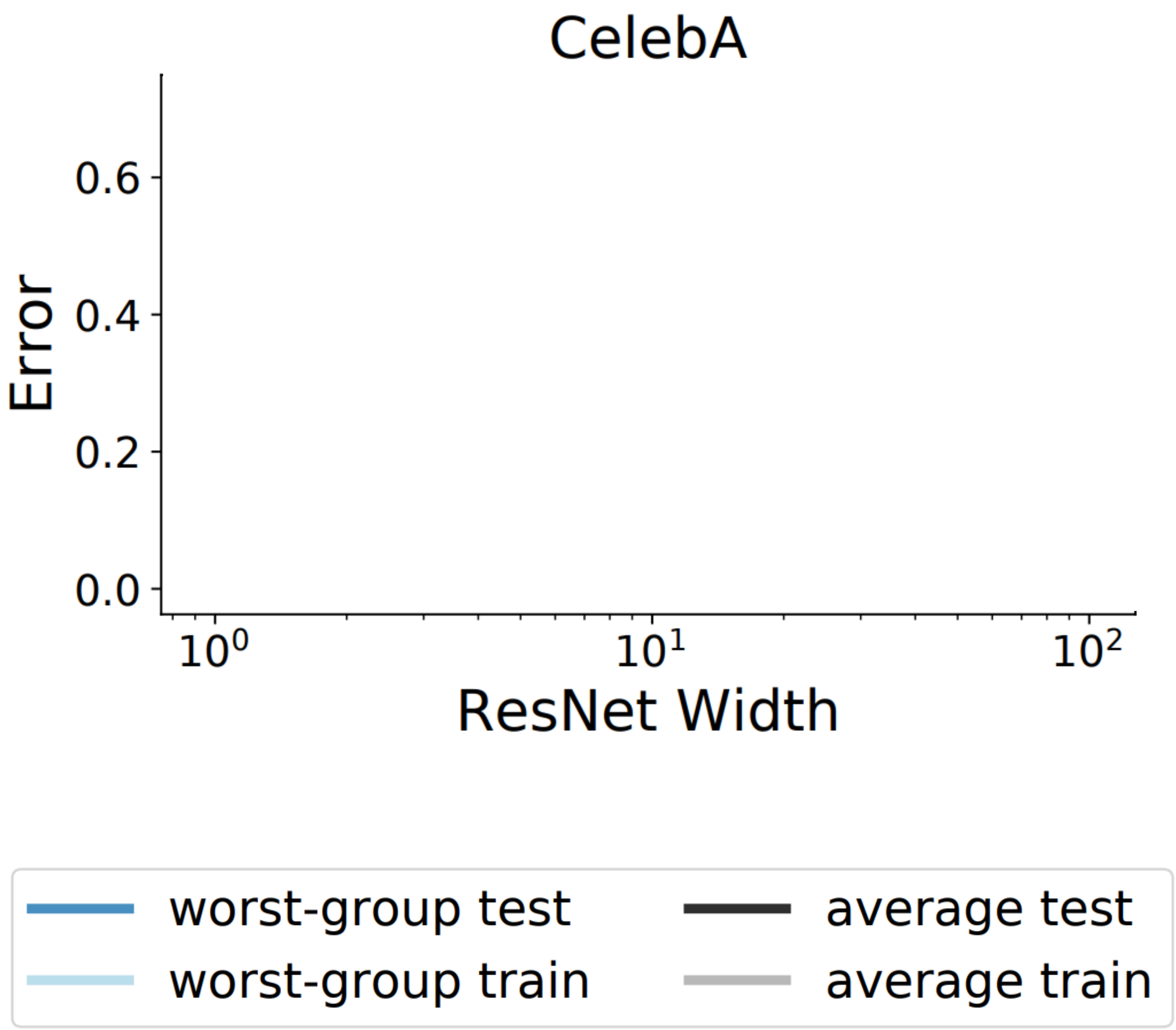
(Sagawa et al. 2020)

- P_{train} is an imbalanced mixture of the groups
- P_{test} is an uniform mixture over all groups

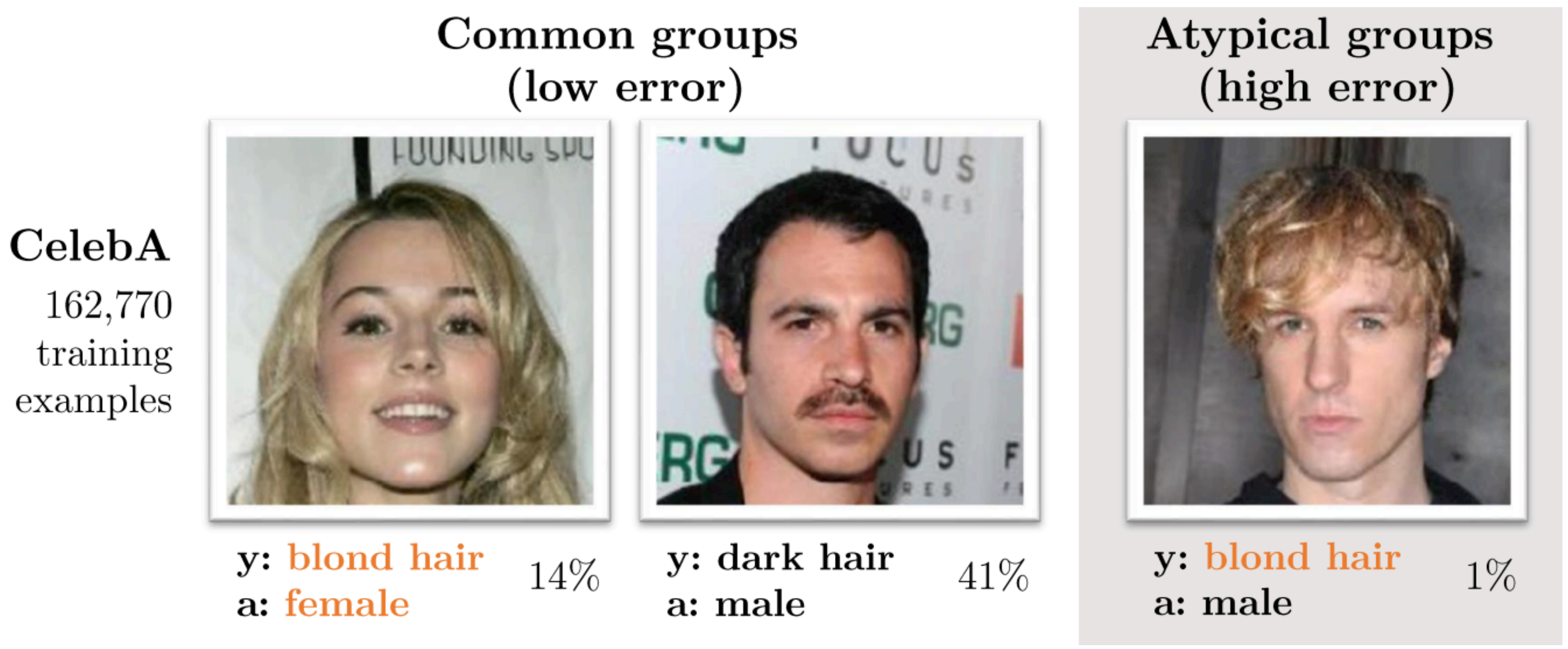
Is Interpolating at odds with Robustness?



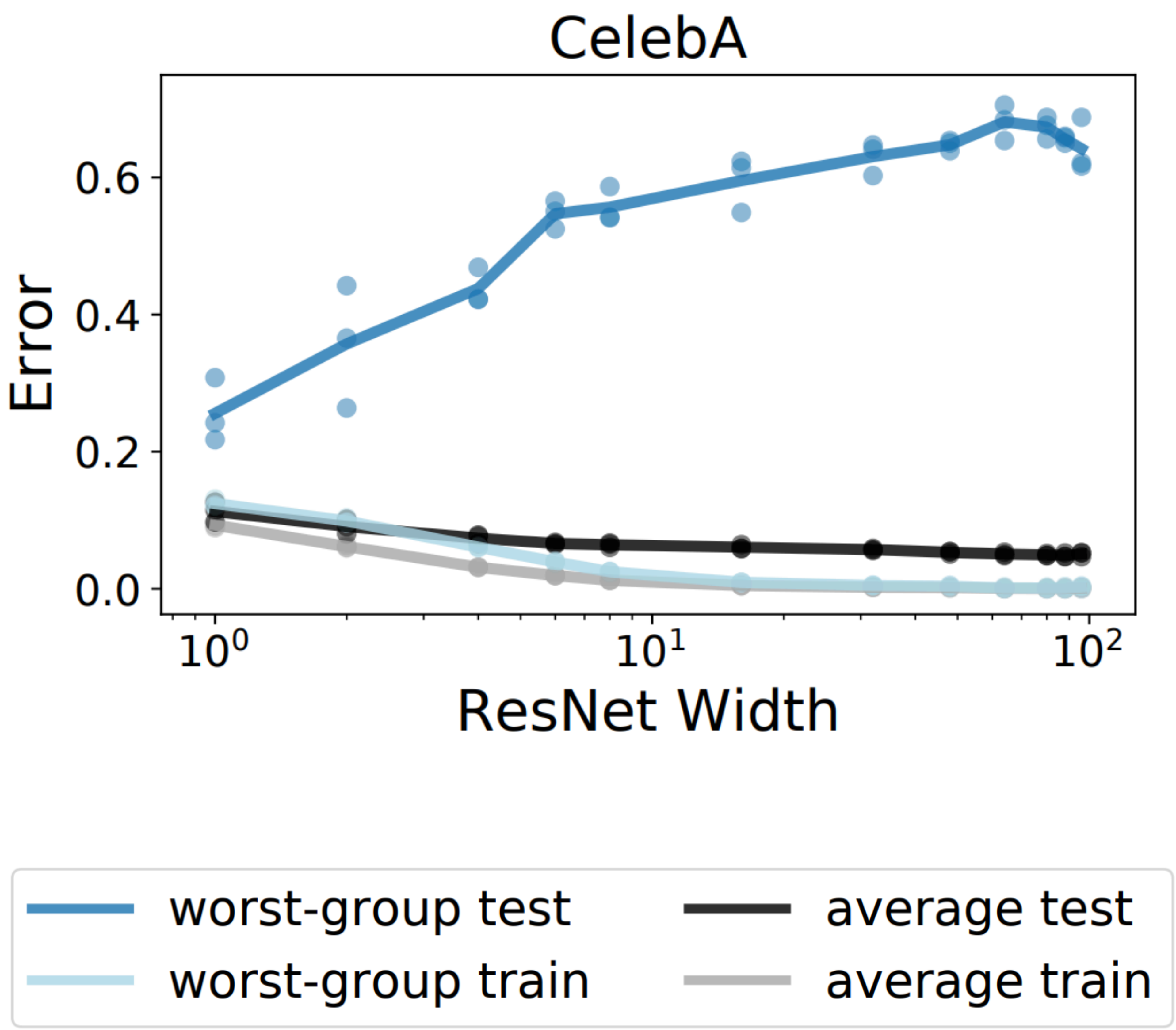
(Sagawa et al. 2020)



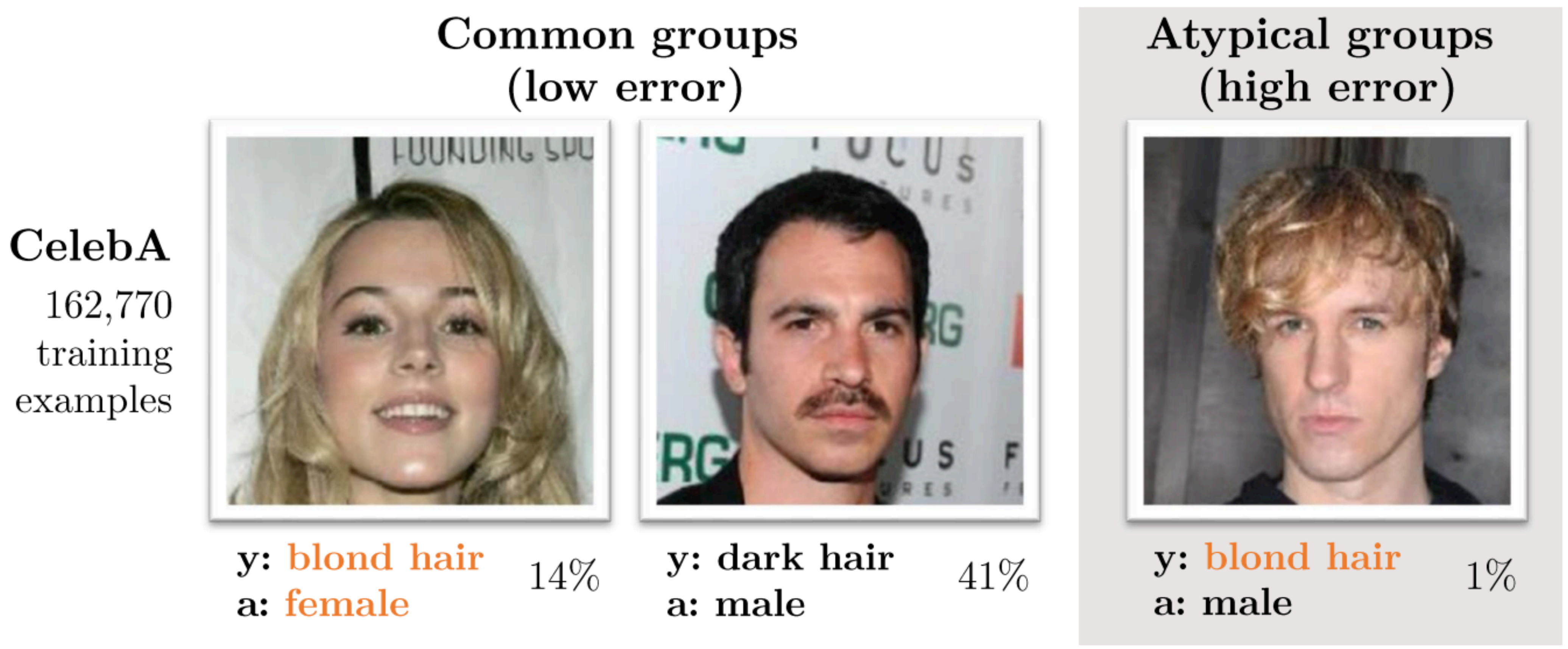
Is Interpolating at odds with Robustness?



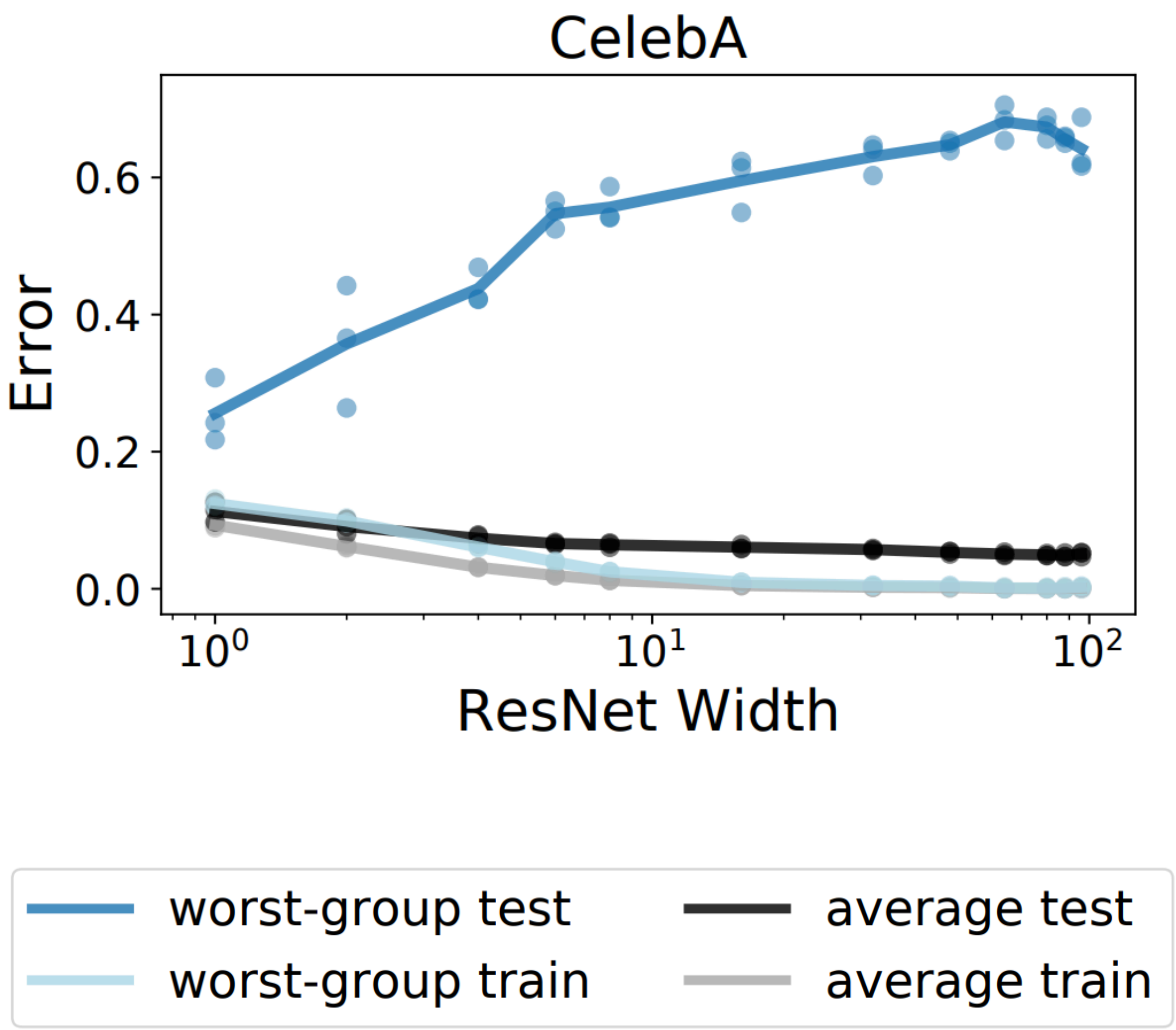
(Sagawa et al. 2020)



Is Interpolating at odds with Robustness?



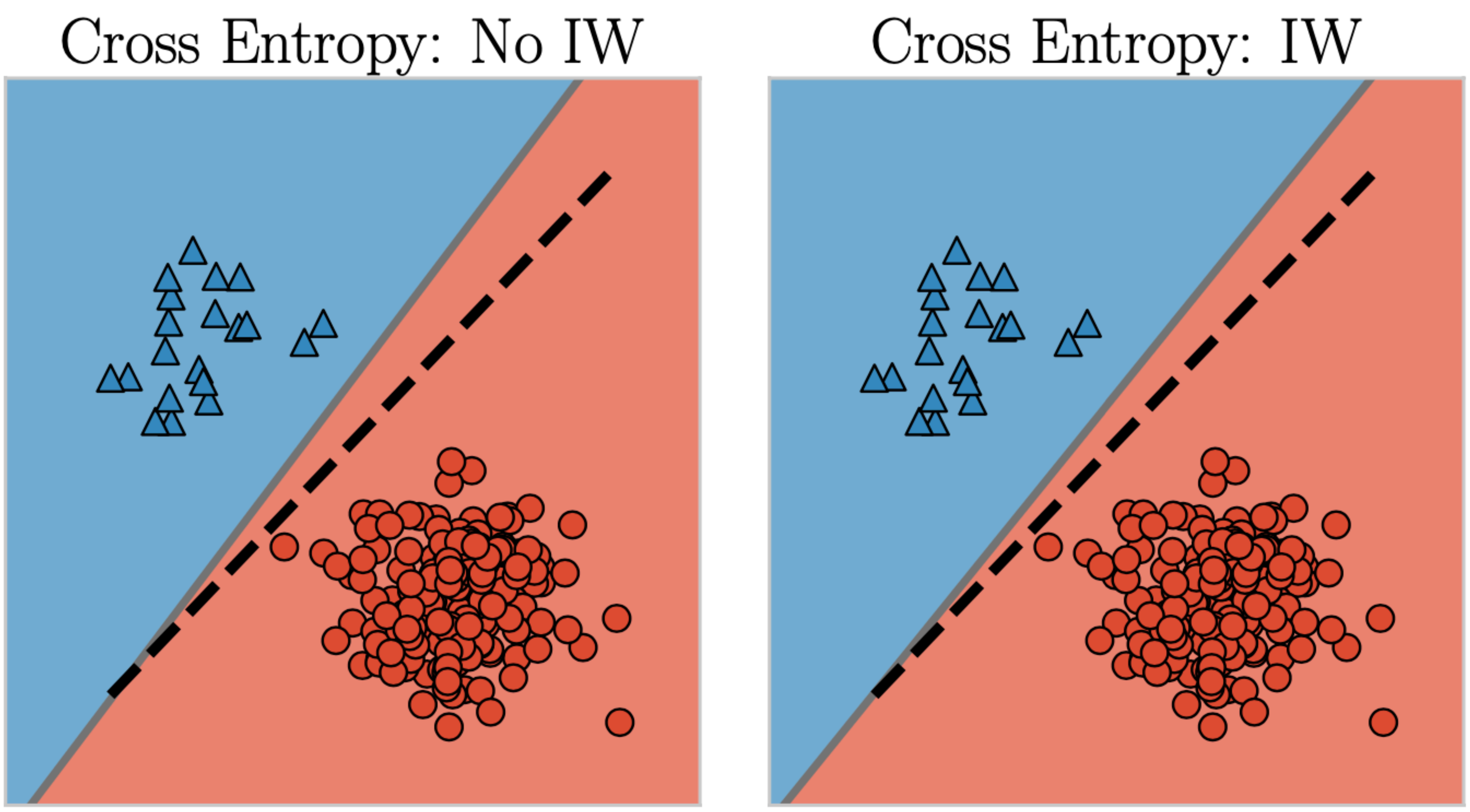
(Sagawa et al. 2020)



Interpolating classifiers trained on the reweighted CE loss suffer high test error

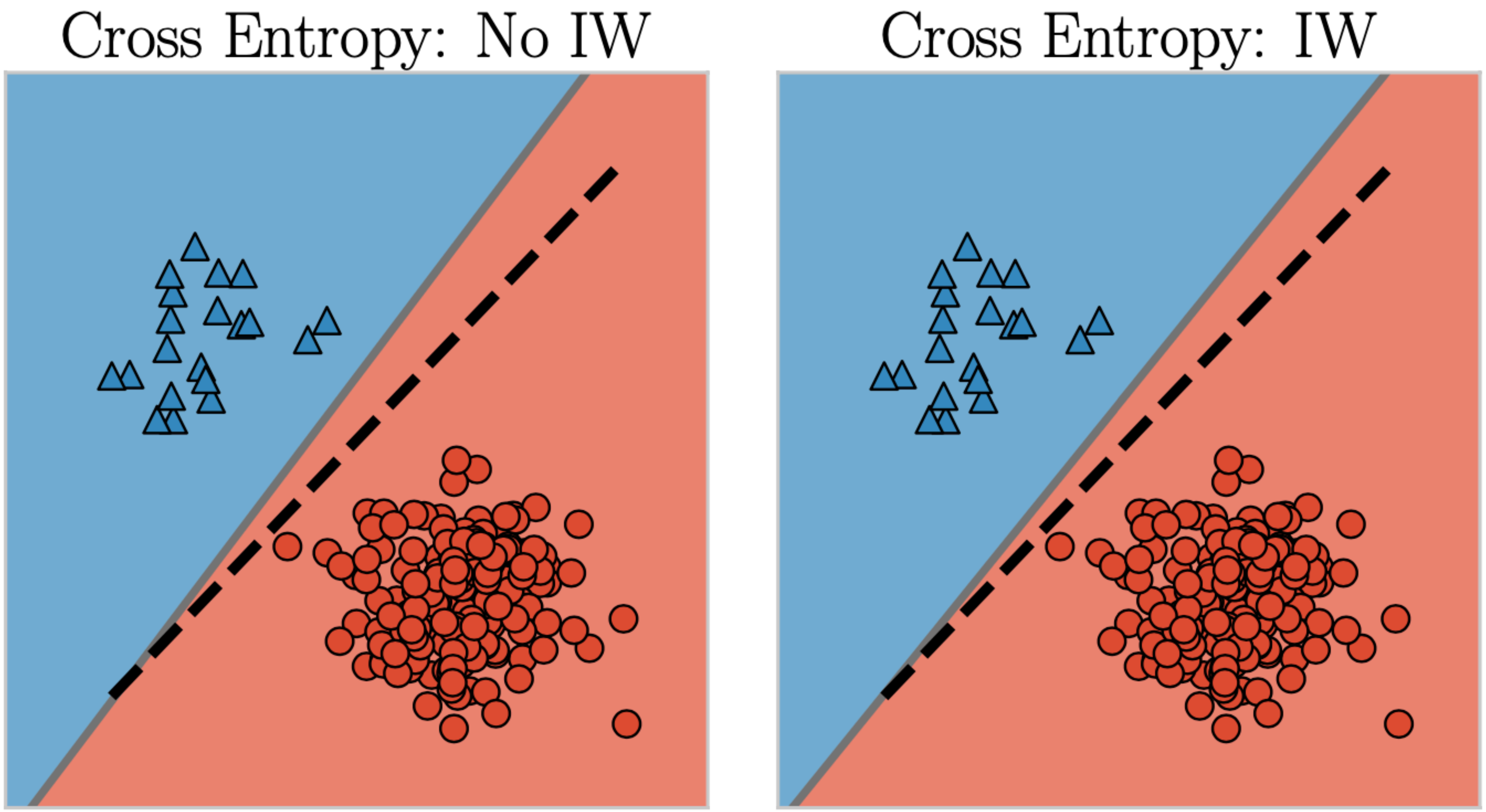
Interpolation breaks Robustness Interventions

Training dynamics of a linear classifier with 2D toy data



Interpolation breaks Robustness Interventions

Training dynamics of a linear classifier with 2D toy data

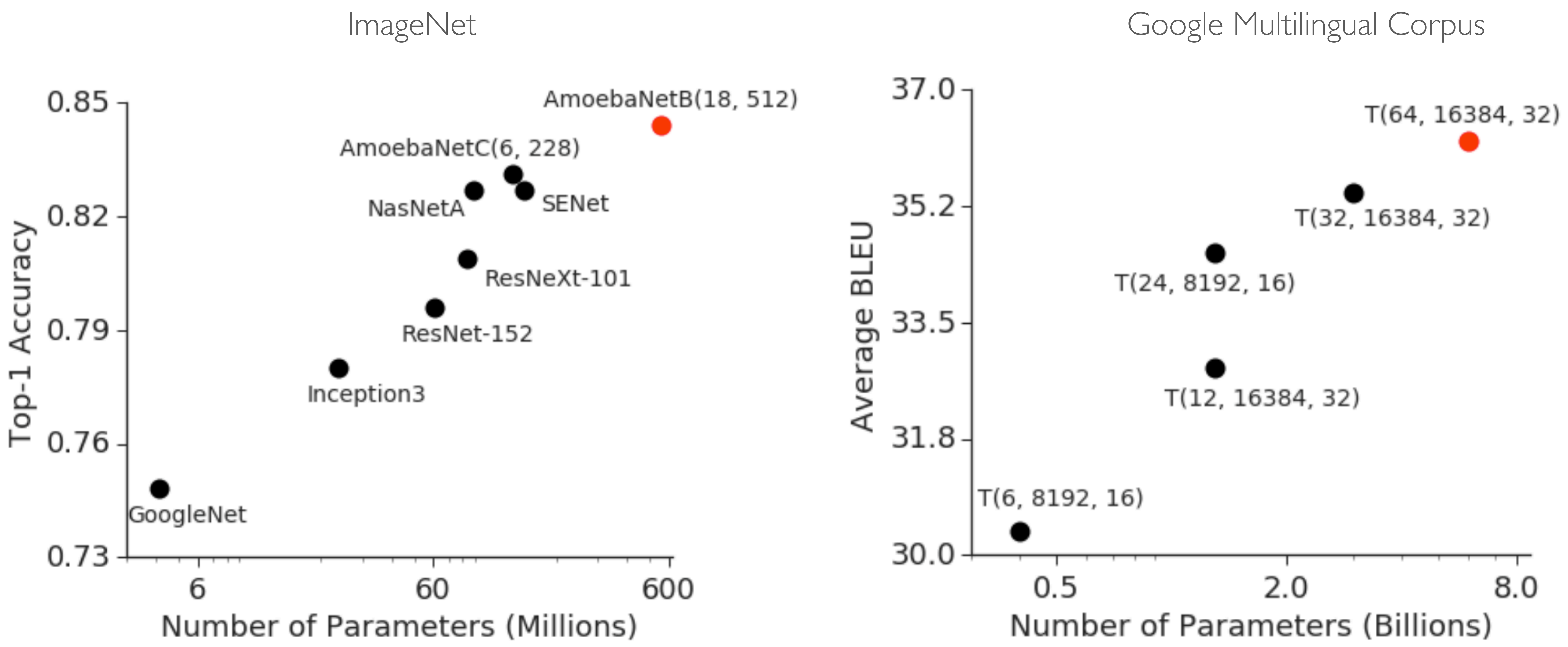


Reweighting results in identical interpolating classifiers!

Is Reweighting Incompatible with Interpolation?

Vignette II: Training Sparse Models

Scaling model size has led to drastic improvements



(Huang et al. 2019)

Vignette II: Training Sparse Models

Scaling model size has led to drastic improvements

However, increased memory and inference time

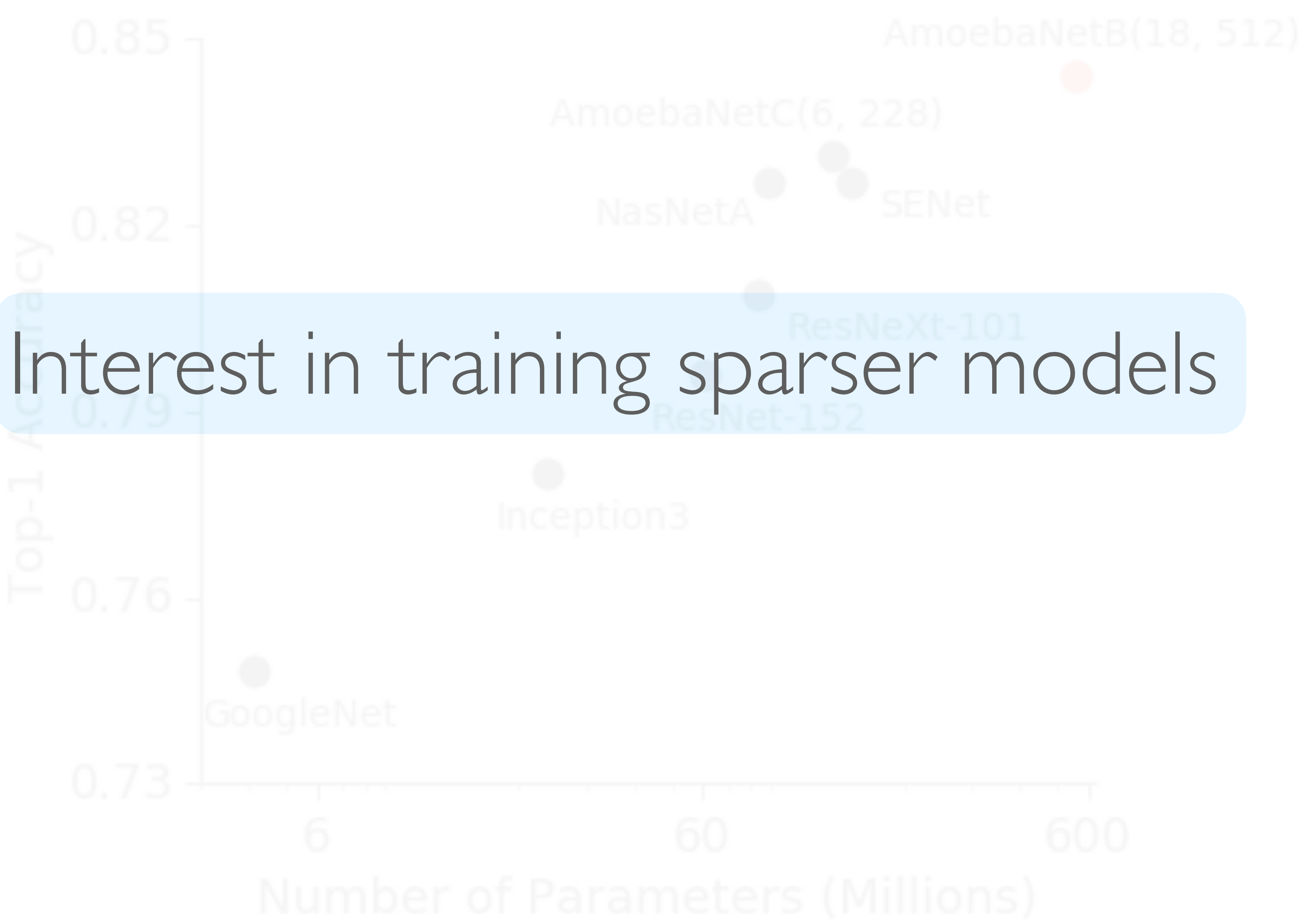
(Huang et al. 2019)

Vignette II: Training Sparse Models

Scaling model size has led to drastic improvements

However, increased memory and inference time

Interest in training sparser models



(Huang et al. 2019)

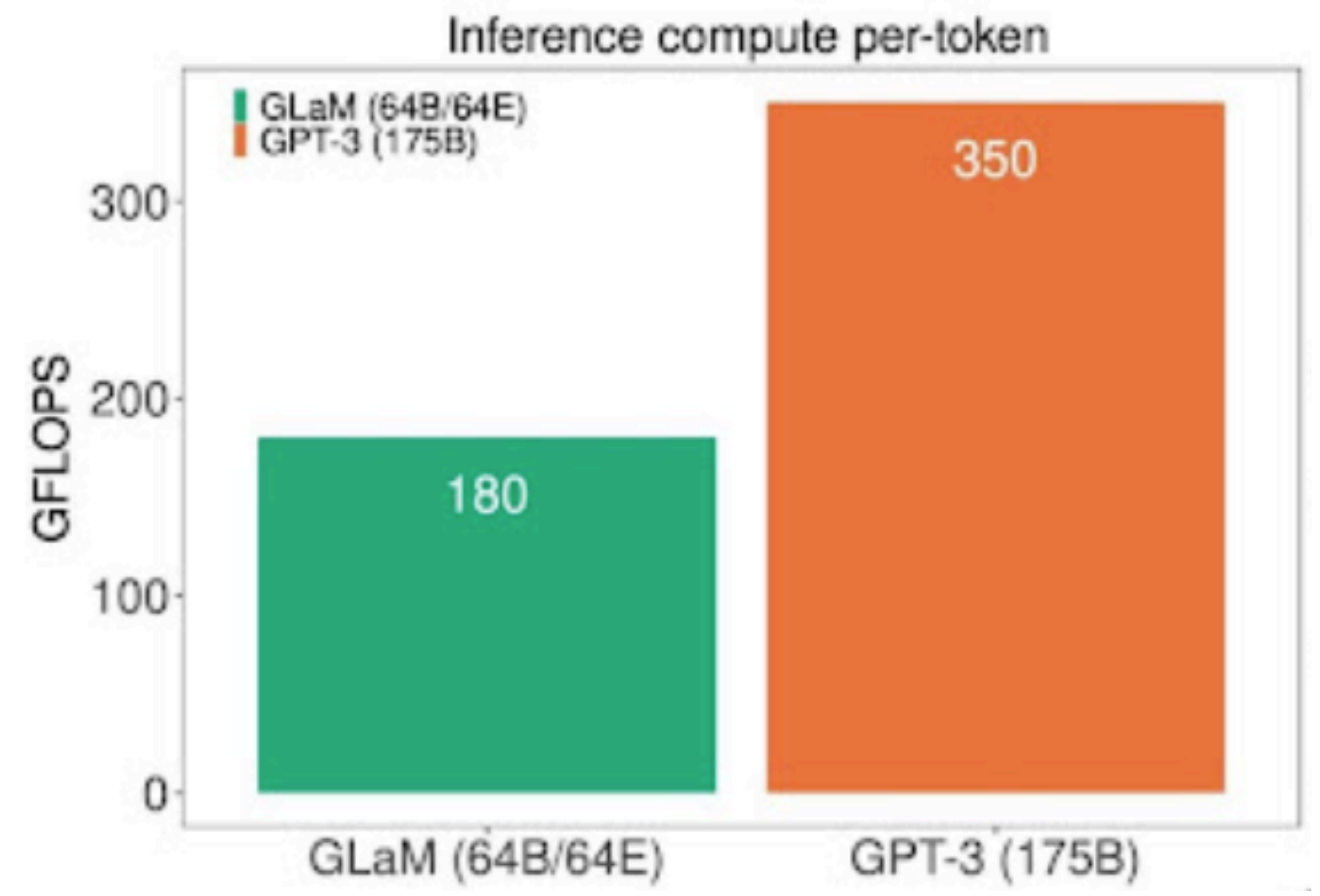
Example: Large Language Models

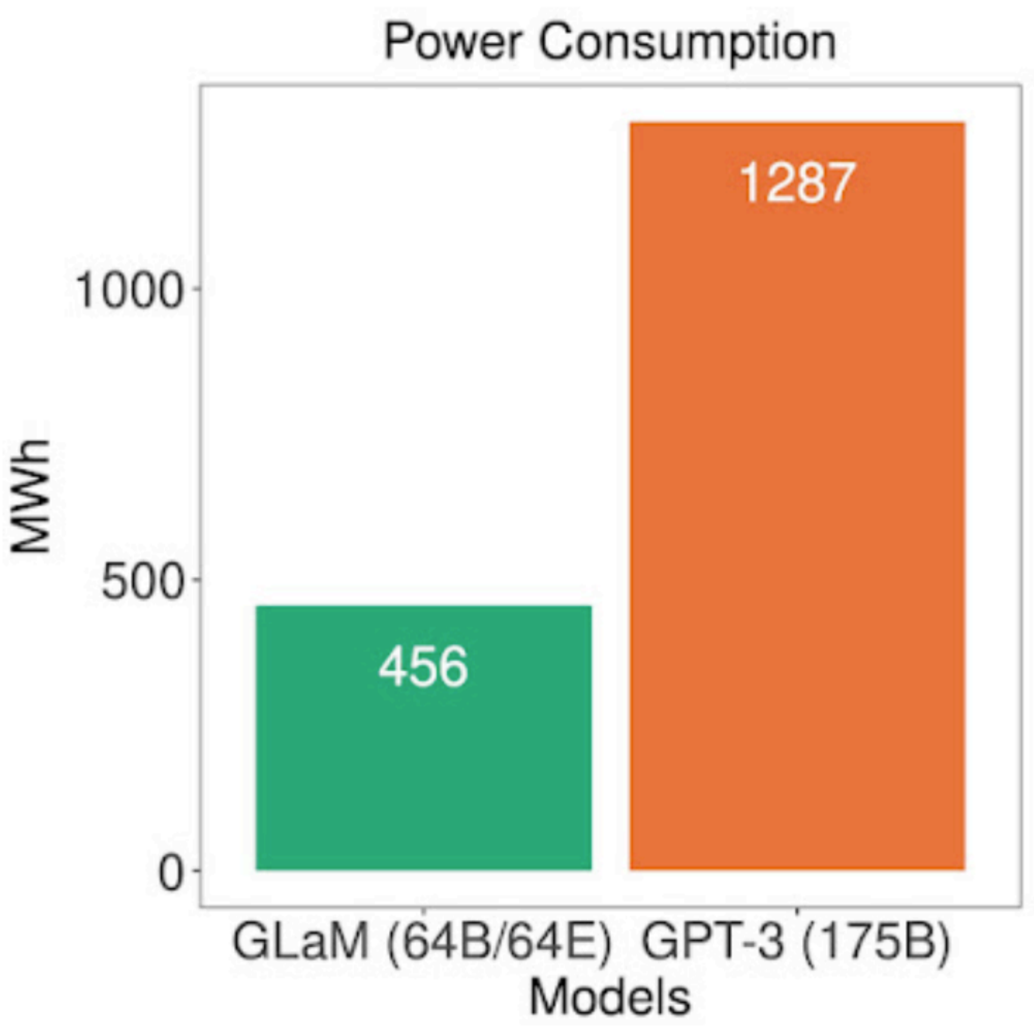
Example: Large Language Models

To speed up inference and efficiency, *sparse* mixture-of-experts models

Example: Large Language Models

To speed up inference and efficiency, *sparse* mixture-of-experts models

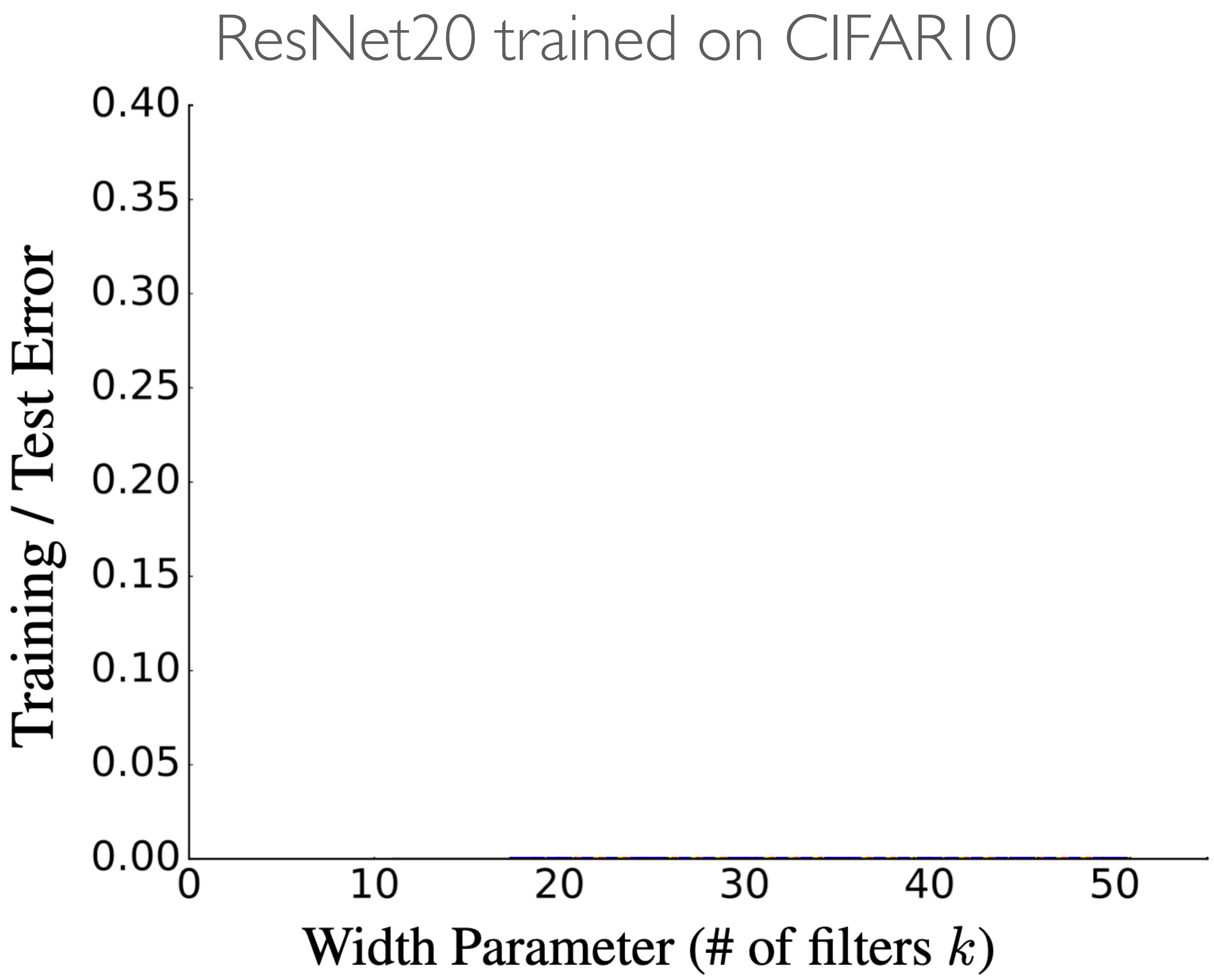




(Dai et al. 2022)

However Sparsity can hurt test error

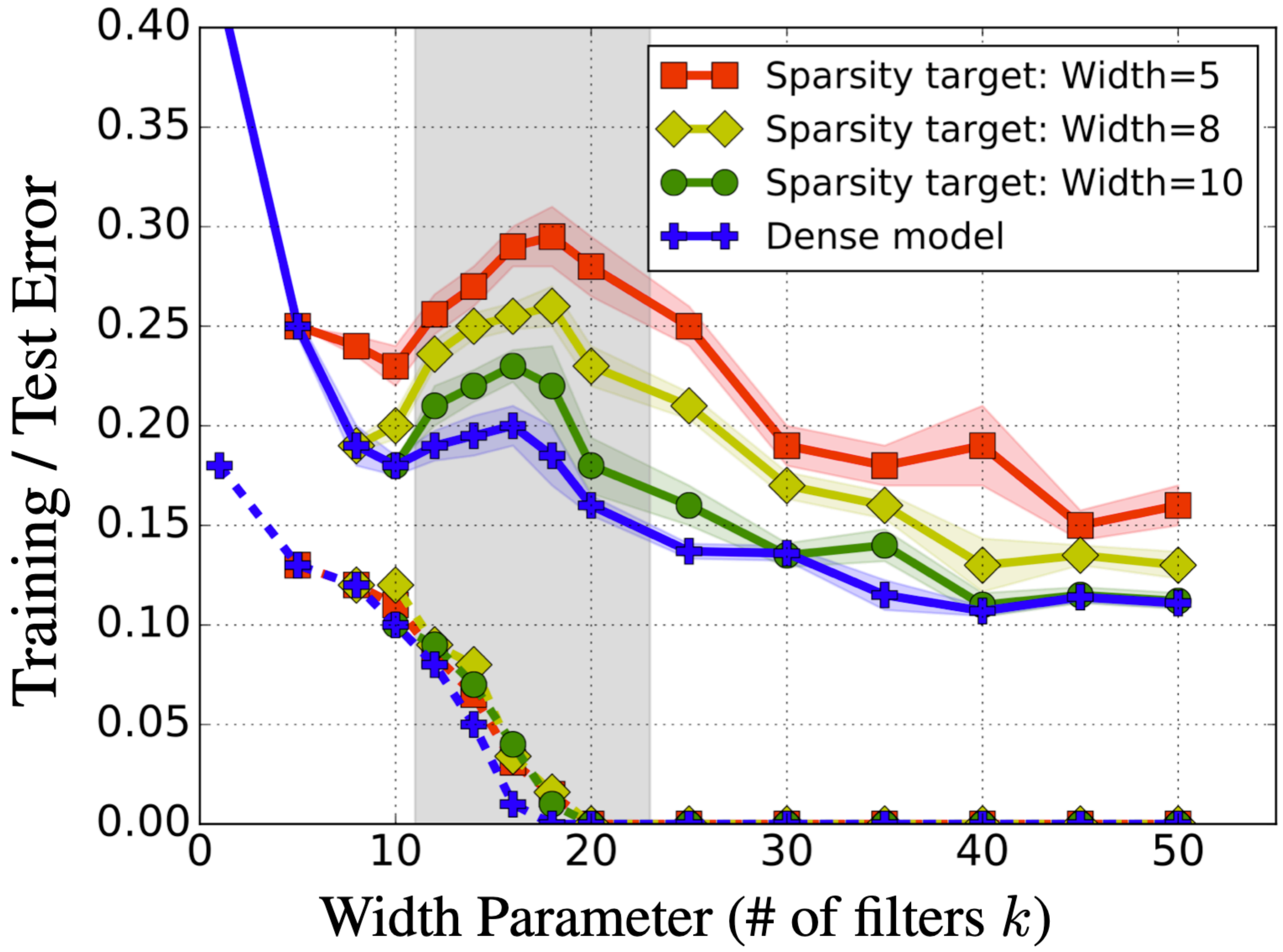
(Chan et al. 2021)



However Sparsity can hurt test error

ResNet20 trained on CIFAR10

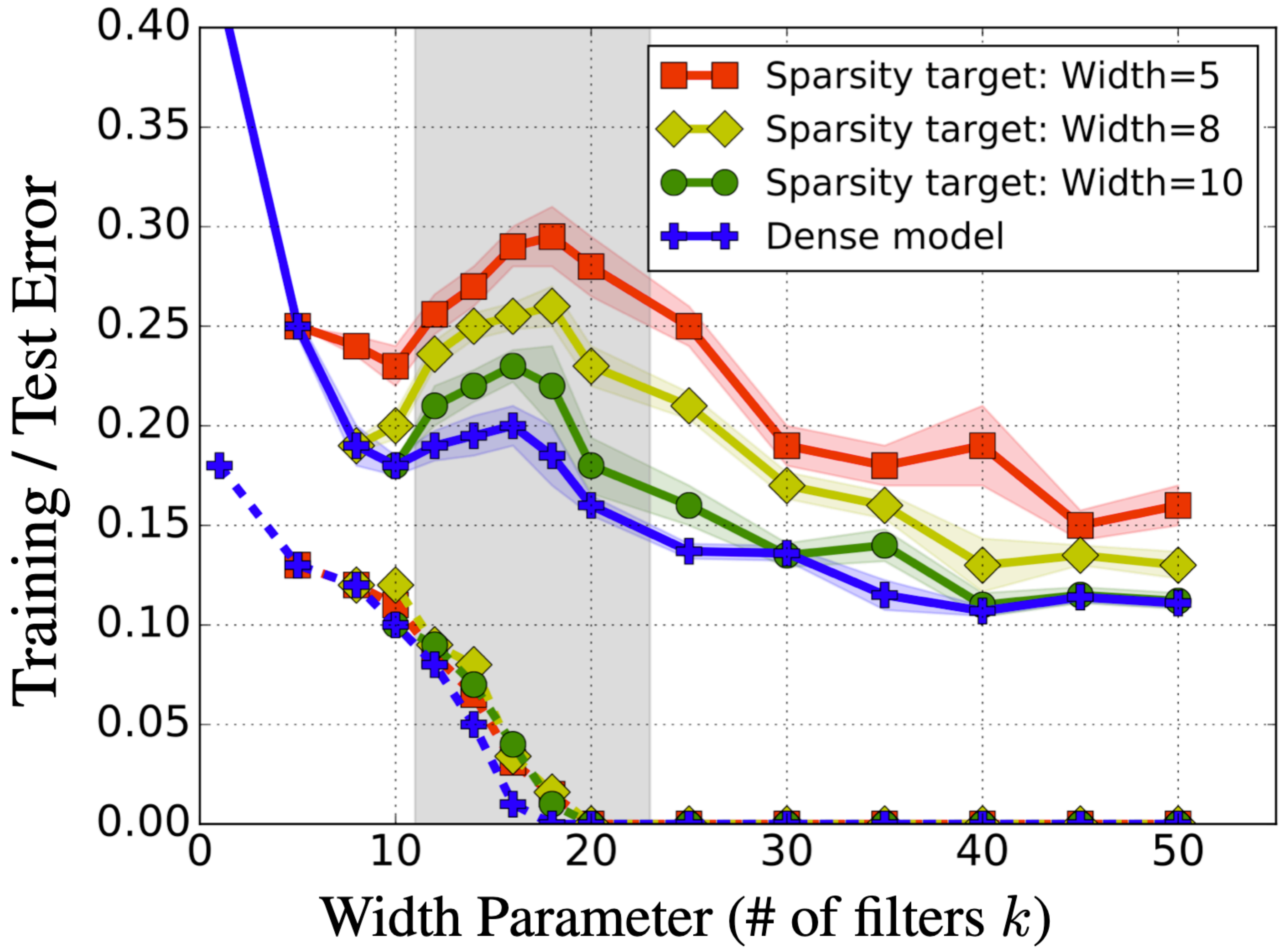
(Chan et al. 2021)



However Sparsity can hurt test error

ResNet20 trained on CIFAR10

(Chan et al. 2021)



As sparsity increases, the test error degrades

Is Sparsity Incompatible with Interpolation?

Talk Outline

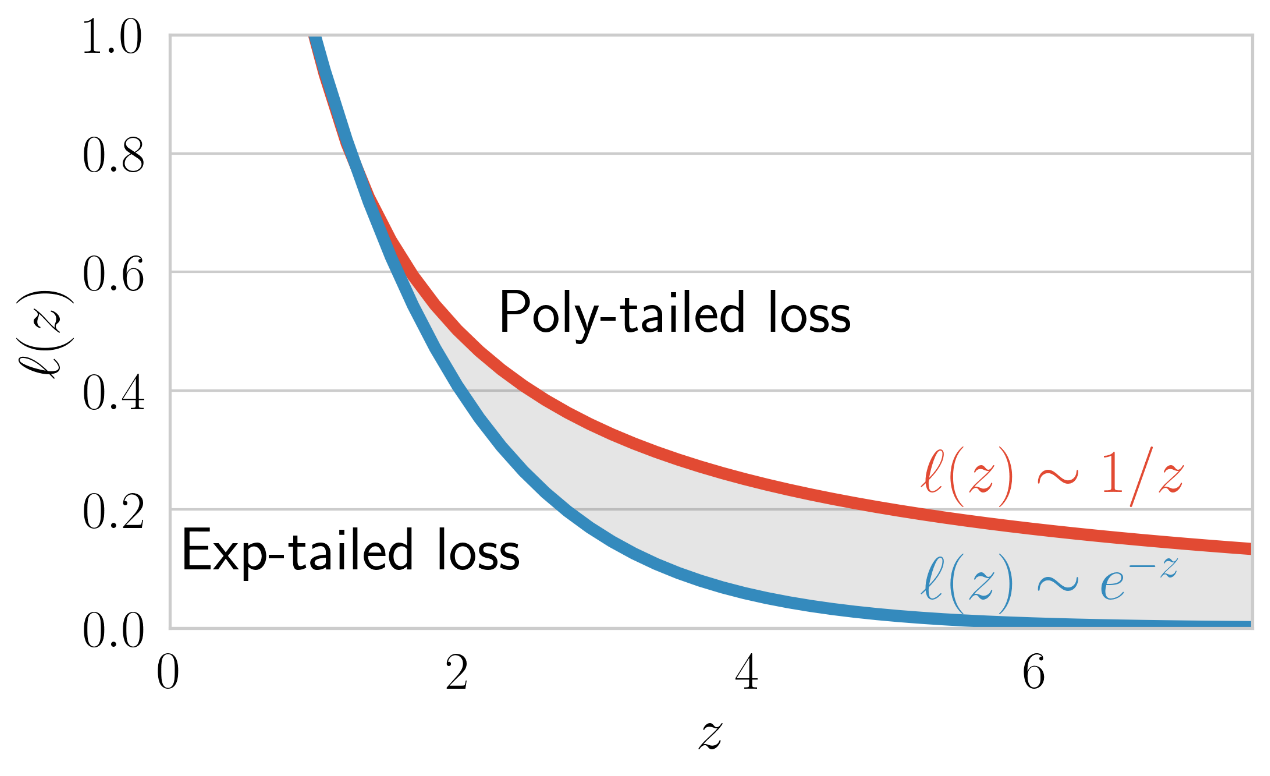
Talk Outline

Study these non-standard settings with linear models

Talk Outline

Study these non-standard settings with linear models

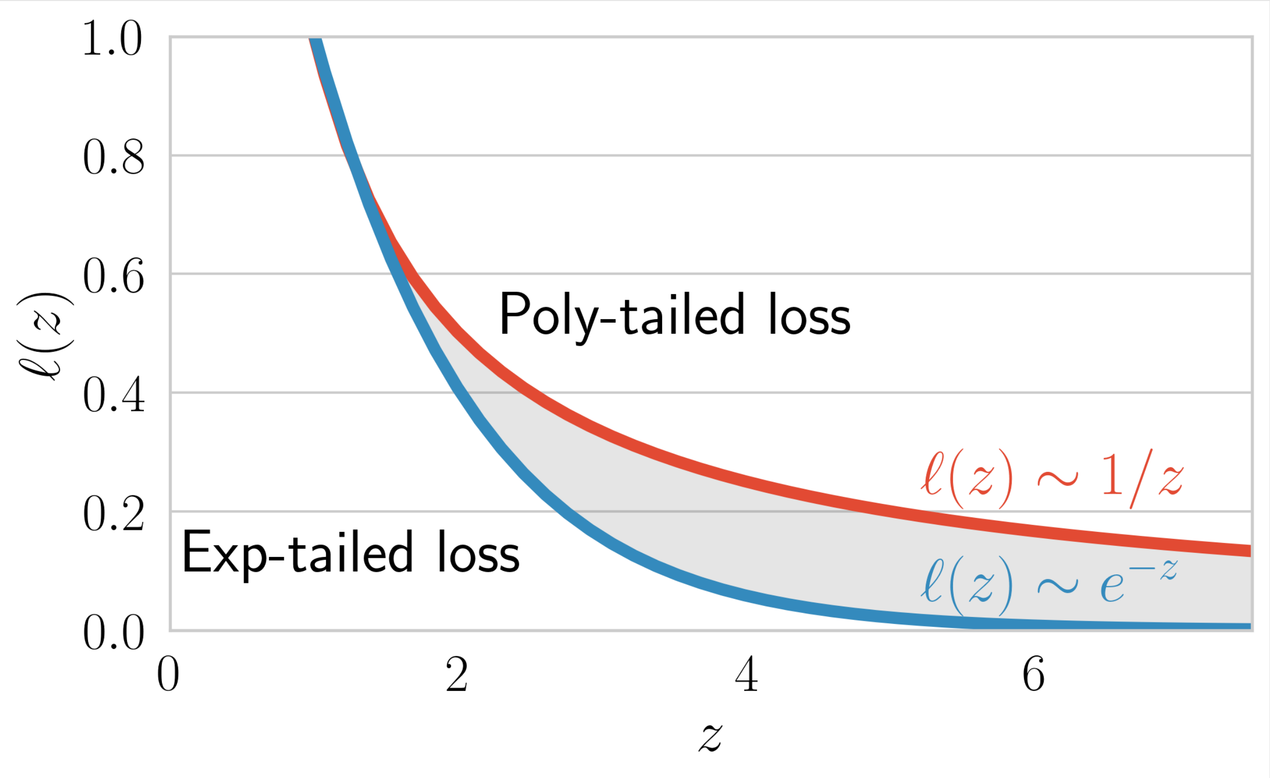
Interpolation under Distribution Shift



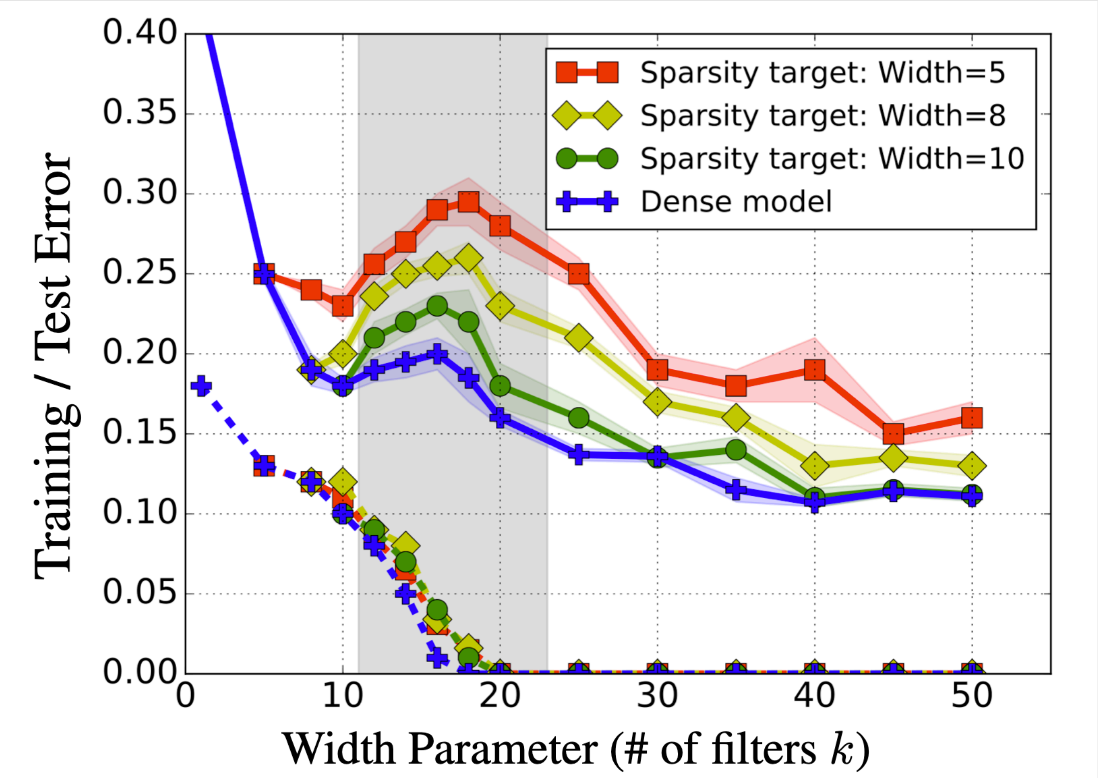
Talk Outline

Study these non-standard settings with linear models

Interpolation under Distribution Shift



Sparsity and Interpolation



Importance Weighting with Interpolating Classifiers

Classification under Distribution Shift

Consider a binary classification task with distribution shift

Classification under Distribution Shift

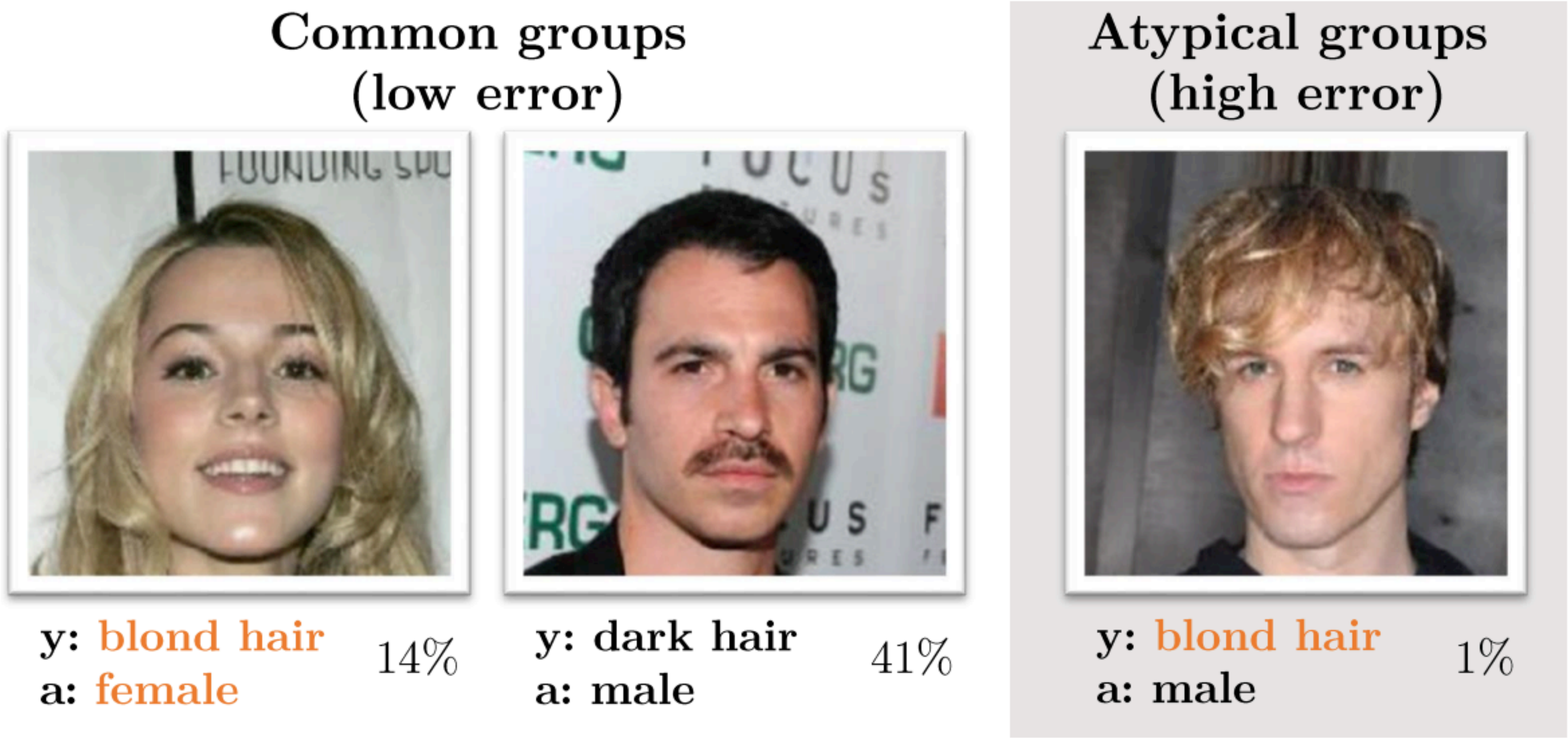
Consider a binary classification task with distribution shift

Given data $(x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^d \times \{-1,1\} \sim \mathsf{P}_{\text{train}}$

Classification under Distribution Shift

Consider a binary classification task with distribution shift

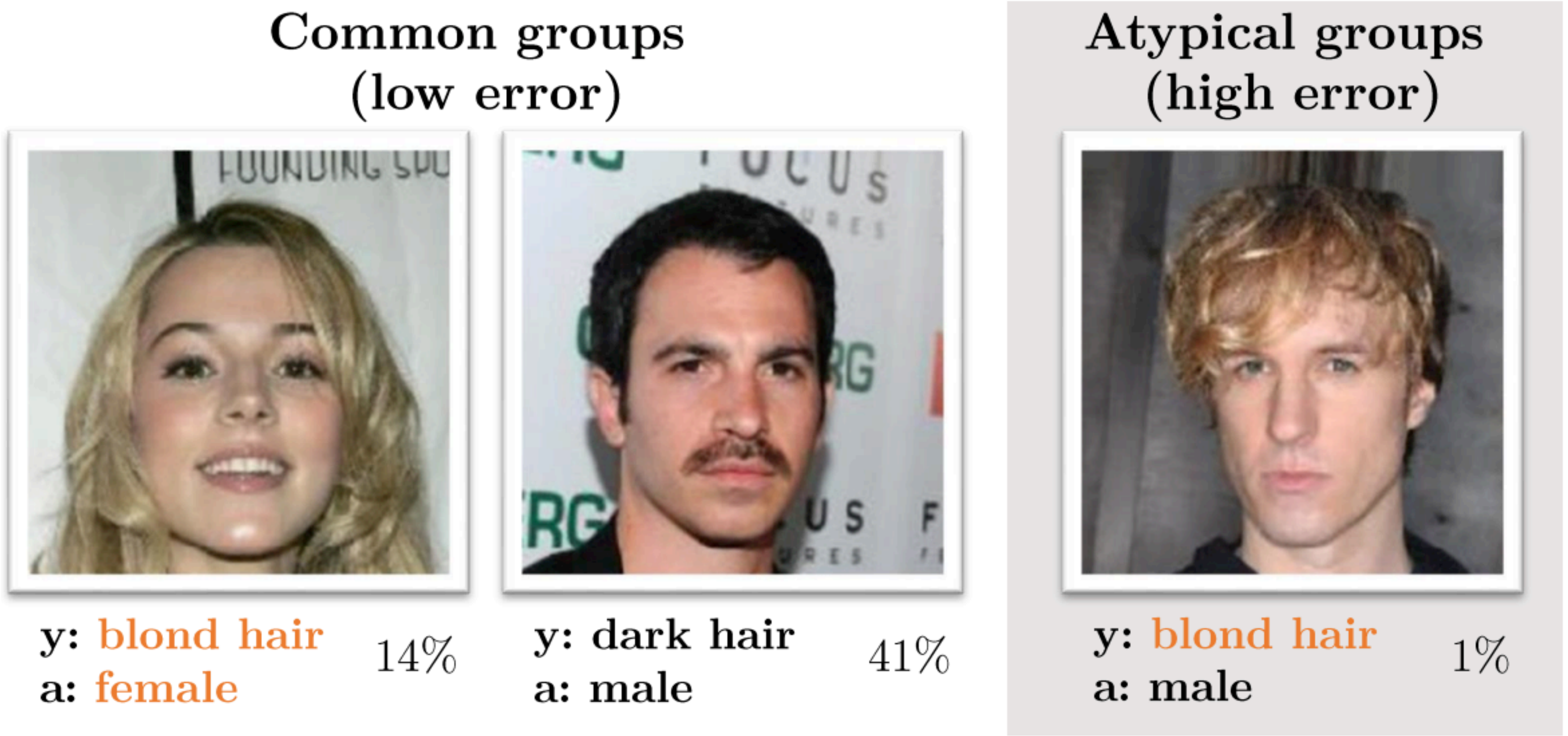
Given data $(x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^d \times \{-1,1\} \sim \mathrm{P}_{\mathrm{train}}$



Classification under Distribution Shift

Consider a binary classification task with distribution shift

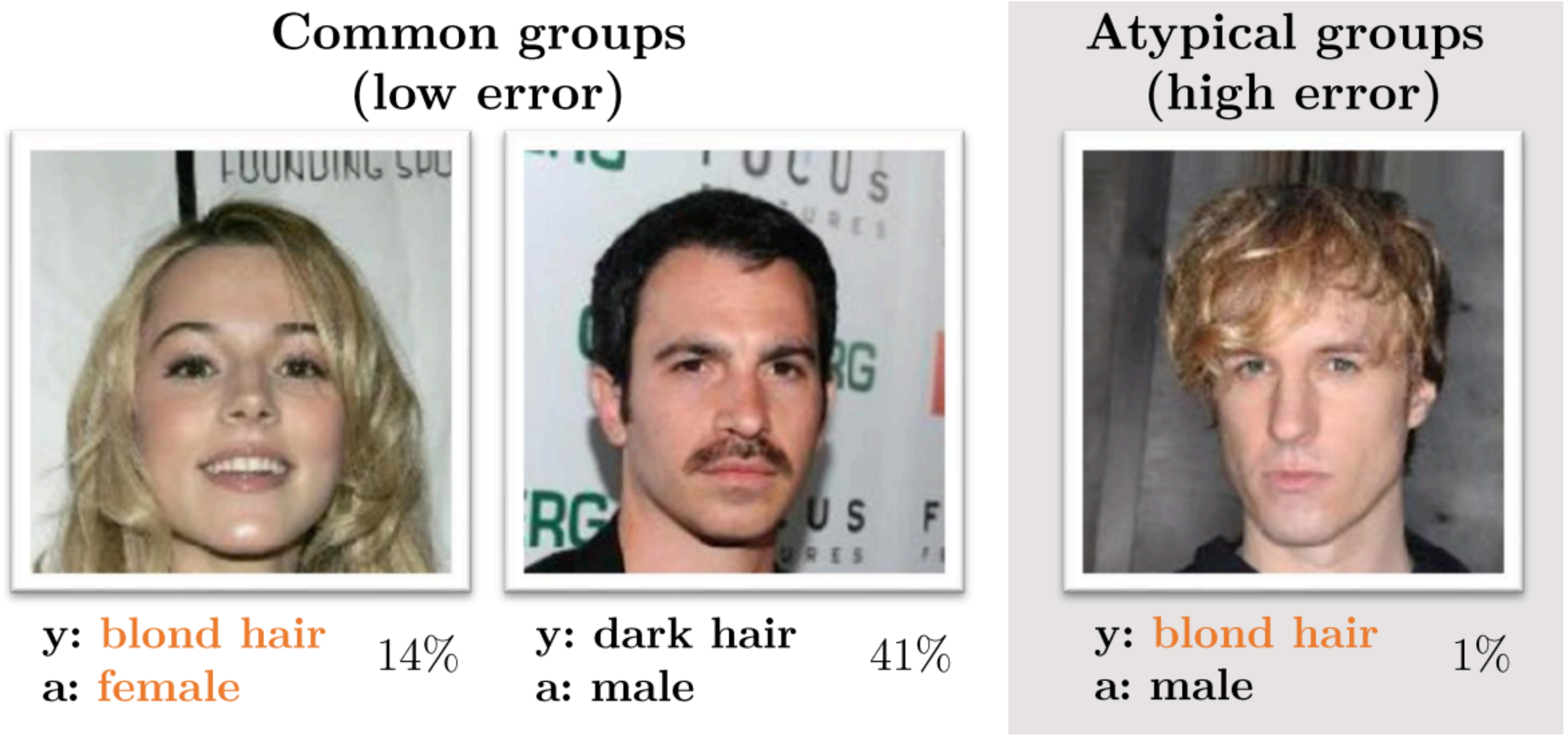
Given data $(x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^d \times \{-1,1\} \sim \mathsf{P}_{\text{train}}$



Goal: minimize test error $\mathbb{P}_{(x,y)\sim \mathsf{P}_{\text{test}}}\left[f_\theta(x) \neq y\right]$

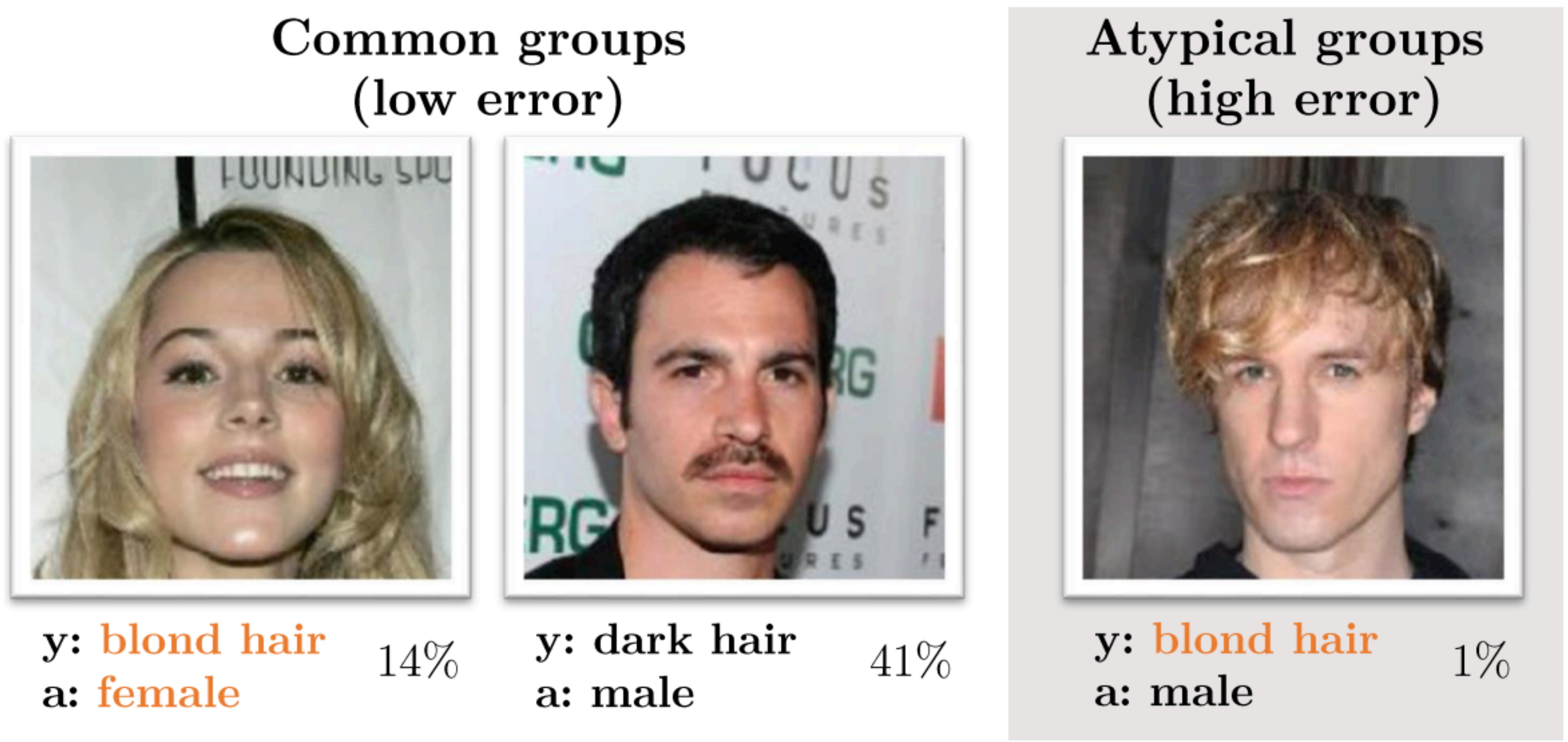
Distribution Shift and Importance Weighting

Goal: minimize test error $\mathbb{P}_{(x,y)\sim P_{\text{test}}}[f_\theta(x) \neq y]$



Distribution Shift and Importance Weighting

Goal: minimize test error $\mathbb{P}_{(x,y)\sim P_{\text{test}}}\left[f_\theta(x) \neq y\right]$

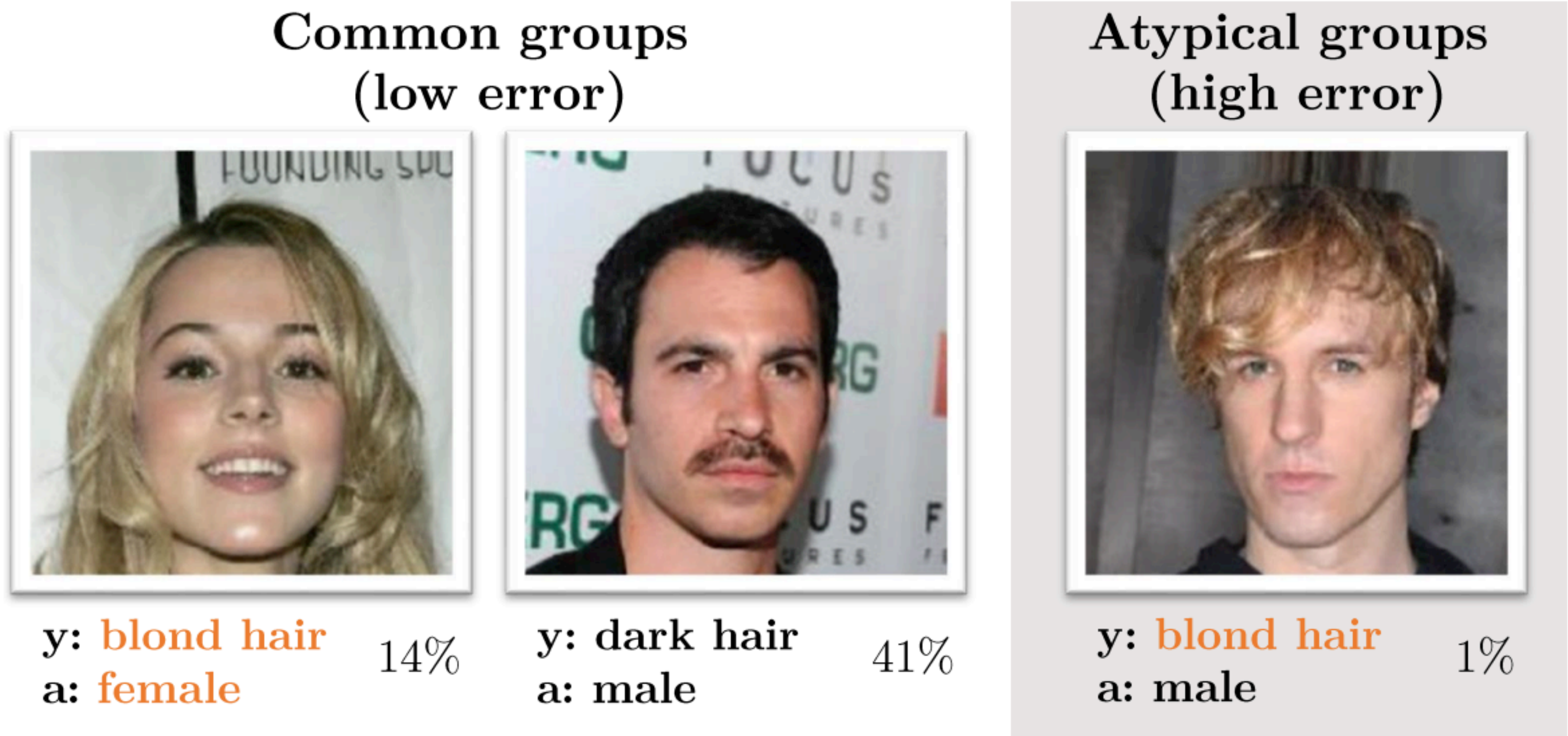


Use gradient descent to minimize the *importance weighted* loss (Shimodaira 2000)

$$L(f(\theta)) = \sum_{i=1}^{n} w_i \log\left[1 + \exp(-y_i f_\theta(x))\right]$$

Distribution Shift and Importance Weighting

Goal: minimize test error $\mathbb{P}_{(x,y)\sim \mathrm{P}_{\mathrm{test}}}[f_\theta(x) \neq y]$



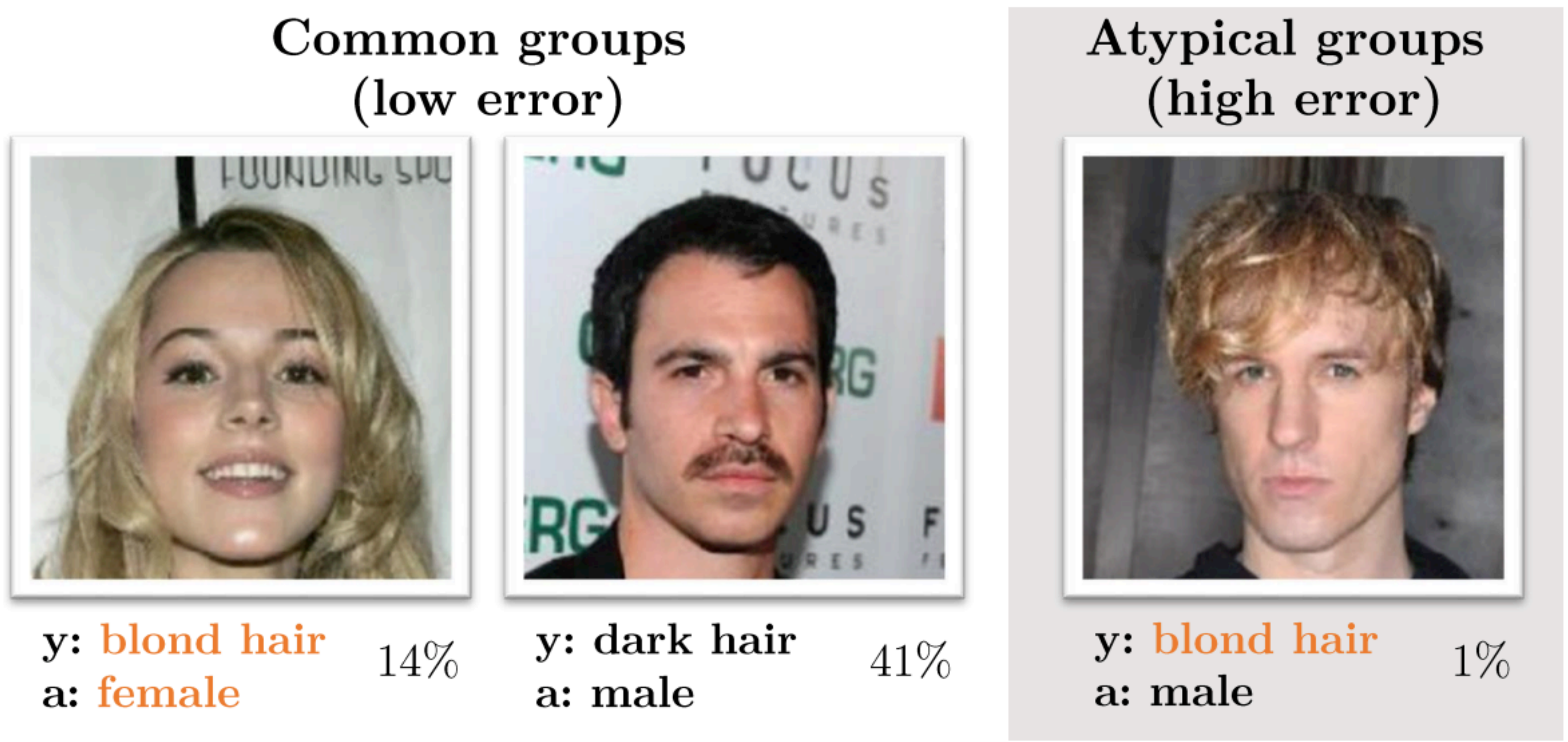
Use gradient descent to minimize the *importance weighted* loss (Shimodaira 2000)

$$L(f(\theta)) = \sum_{i=1}^{n} w_i \log\left[1 + \exp(-y_i f_\theta(x))\right]$$

Standard choice $w_i = \dfrac{\mathrm{P}_{\mathrm{test}}(x_i, y_i)}{\mathrm{P}_{\mathrm{train}}(x_i, y_i)}$

Distribution Shift and Importance Weighting

Goal: minimize test error $\mathbb{P}_{(x,y)\sim \mathsf{P}_{\text{test}}}\left[f_\theta(x) \neq y\right]$



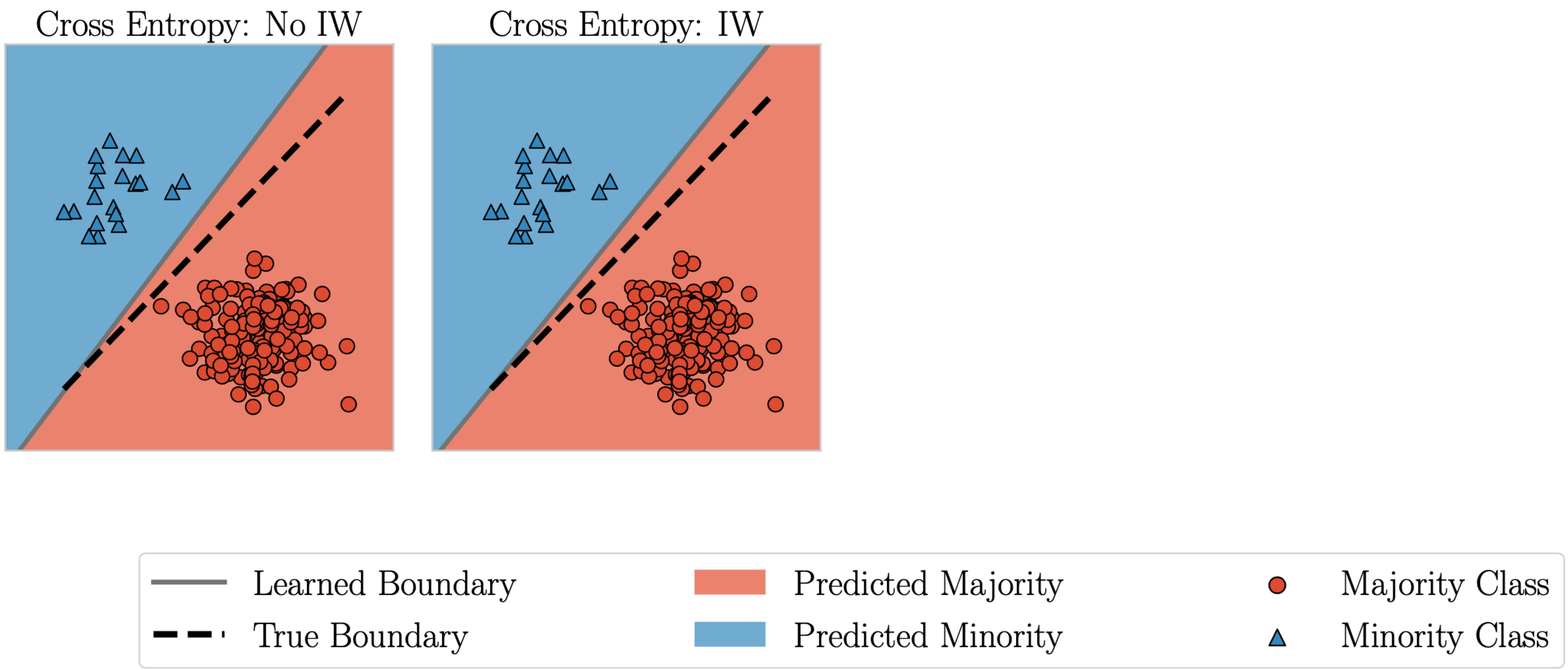
Use gradient descent to minimize the *importance weighted* loss (Shimodaira 2000)

$$L(f(\theta)) = \sum_{i=1}^{n} w_i \log\left[1 + \exp(-y_i f_\theta(x))\right]$$

Standard choice $w_i = \dfrac{\mathsf{P}_{\text{test}}(x_i, y_i)}{\mathsf{P}_{\text{train}}(x_i, y_i)}$

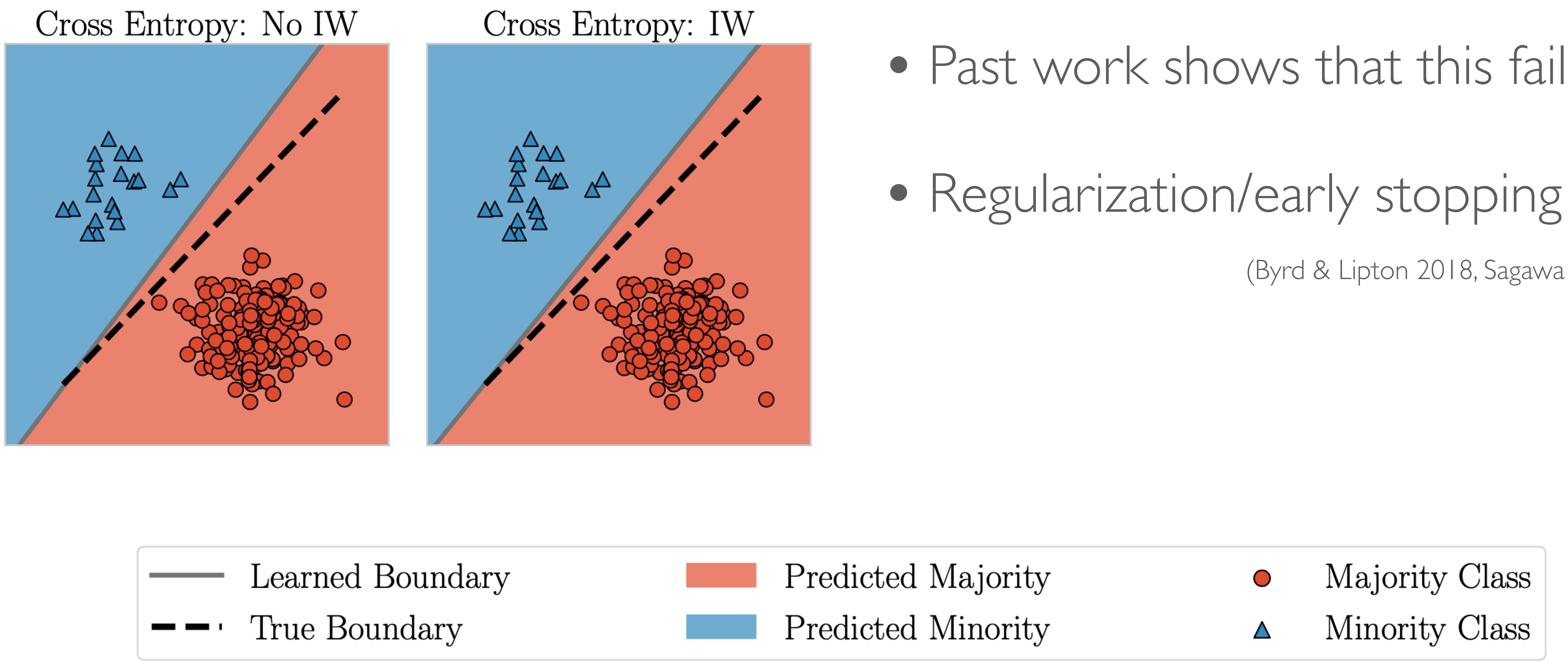
Train until interpolation: $L(f(\theta^{(t)})) \to 0$

Doesn't work



Cross Entropy: No IW
Cross Entropy: IW
Learned Boundary
True Boundary
Predicted Majority
Predicted Minority
Majority Class
Minority Class

Doesn’t work



- Past work shows that this fails
- Regularization/early stopping helps

(Byrd & Lipton 2018, Sagawa et al. 2019)

Can we design interpolators that respond to weighting?

Why do exp-tailed losses fail?

Why do exp-tailed losses fail?

Given linearly separable data $(x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^d \times \{-1,1\} \sim \mathrm{P}_{\text{train}}$

If gradient descent used to minimize

$$L(\theta) = \sum_{i=1}^{n} w_i \log\left[1 + \exp(-y_i x_i^\top \theta)\right]$$

Why do exp-tailed losses fail?

Given linearly separable data $(x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^d \times \{-1, 1\} \sim \mathsf{P}_{\text{train}}$

If gradient descent used to minimize

$$L(\theta) = \sum_{i=1}^{n} w_i \log\left[1 + \exp(-y_i x_i^\top \theta)\right]$$

(Xu et al. 2019, Soudry et al. 2018, Ji and Telgarsky 2018)

Given iterates $\theta^{(t+1)} = \theta^{(t)} - \eta \nabla L(\theta^{(t)})$ if η is small enough

Why do exp-tailed losses fail?

Given linearly separable data $(x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^d \times \{-1,1\} \sim \mathrm{P}_{\text{train}}$

If gradient descent used to minimize

$$L(\theta) = \sum_{i=1}^{n} w_i \log\left[1 + \exp(-y_i x_i^\top \theta)\right]$$

(Xu et al. 2019, Soudry et al. 2018, Ji and Telgarsky 2018)

Given iterates $\theta^{(t+1)} = \theta^{(t)} - \eta \nabla L(\theta^{(t)})$ if η is small enough

$$\frac{\theta^{(t)}}{\|\theta^{(t)}\|} \to \arg\max_{\|\theta\|_2=1} \left\{\gamma \;:\; \text{subject to } y_i x_i^\top \theta \geq \gamma, \forall i \in [n]\right\}$$

Why do exp-tailed losses fail?

Given linearly separable data $(x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^d \times \{-1,1\} \sim \mathsf{P}_{\text{train}}$

If gradient descent used to minimize

$$L(\theta) = \sum_{i=1}^{n} w_i \log\left[1 + \exp(-y_i x_i^\top \theta)\right]$$

(Xu et al. 2019, Soudry et al. 2018, Ji and Telgarsky 2018)

Given iterates $\theta^{(t+1)} = \theta^{(t)} - \eta \nabla L(\theta^{(t)})$ if η is small enough

$$\frac{\theta^{(t)}}{\|\theta^{(t)}\|} \to \boxed{\arg\max_{\|\theta\|_2=1} \left\{\gamma \;:\; \text{subject to } y_i x_i^\top \theta \geq \gamma, \forall i \in [n]\right\}}$$

"maximum-margin classifier" θ_{MM}

Why do exp-tailed losses fail?

Given linearly separable data $(x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^d \times \{-1,1\} \sim \mathsf{P}_{\text{train}}$

If gradient descent used to minimize

$$L(\theta) = \sum_{i=1}^{n} w_i \log\left[1 + \exp(-y_i x_i^\top \theta)\right]$$

(Xu et al. 2019, Soudry et al. 2018, Ji and Telgarsky 2018)

Given iterates $\theta^{(t+1)} = \theta^{(t)} - \eta \nabla L(\theta^{(t)})$ if η is small enough

$$\frac{\theta^{(t)}}{\|\theta^{(t)}\|} \to \boxed{\arg\max_{\|\theta\|_2 = 1} \left\{\gamma \ : \ \text{subject to } y_i x_i^\top \theta \geq \gamma, \forall i \in [n]\right\}}$$

"maximum-margin classifier" θ_{MM}

Intuition: Reweighting doesn't affect Exp-Tailed Losses

Intuition: Reweighting doesn't affect Exp-Tailed Losses

Consider the reweighted objective $L(\theta) = \sum_{i=1}^{n} w_i \log\left[1 + \exp(-y_i x_i^\top \theta)\right]$

Intuition: Reweighting doesn't affect Exp-Tailed Losses

Consider the reweighted objective $L(\theta) = \sum_{i=1}^{n} w_i \log\left[1 + \exp(-y_i x_i^\top \theta)\right]$

This is equivalent to creating a "*new dataset*" with w_i copies of sample $\boldsymbol{i}$

$$\underbrace{(x_i, y_i), \ldots, (x_i, y_i)}_{w_i \text{ times}}$$

Intuition: Reweighting doesn't affect Exp-Tailed Losses

Consider the reweighted objective $L(\theta) = \sum_{i=1}^{n} w_i \log\left[1 + \exp(-y_i x_i^\top \theta)\right]$

This is equivalent to creating a "*new dataset*" with w_i copies of sample $\boldsymbol{i}$

$$\underbrace{(x_i, y_i), \ldots, (x_i, y_i)}_{w_i \text{ times}}$$

The max-margin classifier for this new dataset is *unchanged*

$$\arg\max_{\|\theta\|_2=1} \left\{\gamma \;:\; \text{subject to } y_i x_i^\top \theta \geq \gamma, \forall i \in [n]\right\}$$

Intuition: Reweighting doesn't affect Exp-Tailed Losses

Consider the reweighted objective $L(\theta) = \sum_{i=1}^{n} w_i \log\left[1 + \exp(-y_i x_i^\top \theta)\right]$

This is equivalent to creating a "*new dataset*" with w_i copies of sample $\boldsymbol{i}$

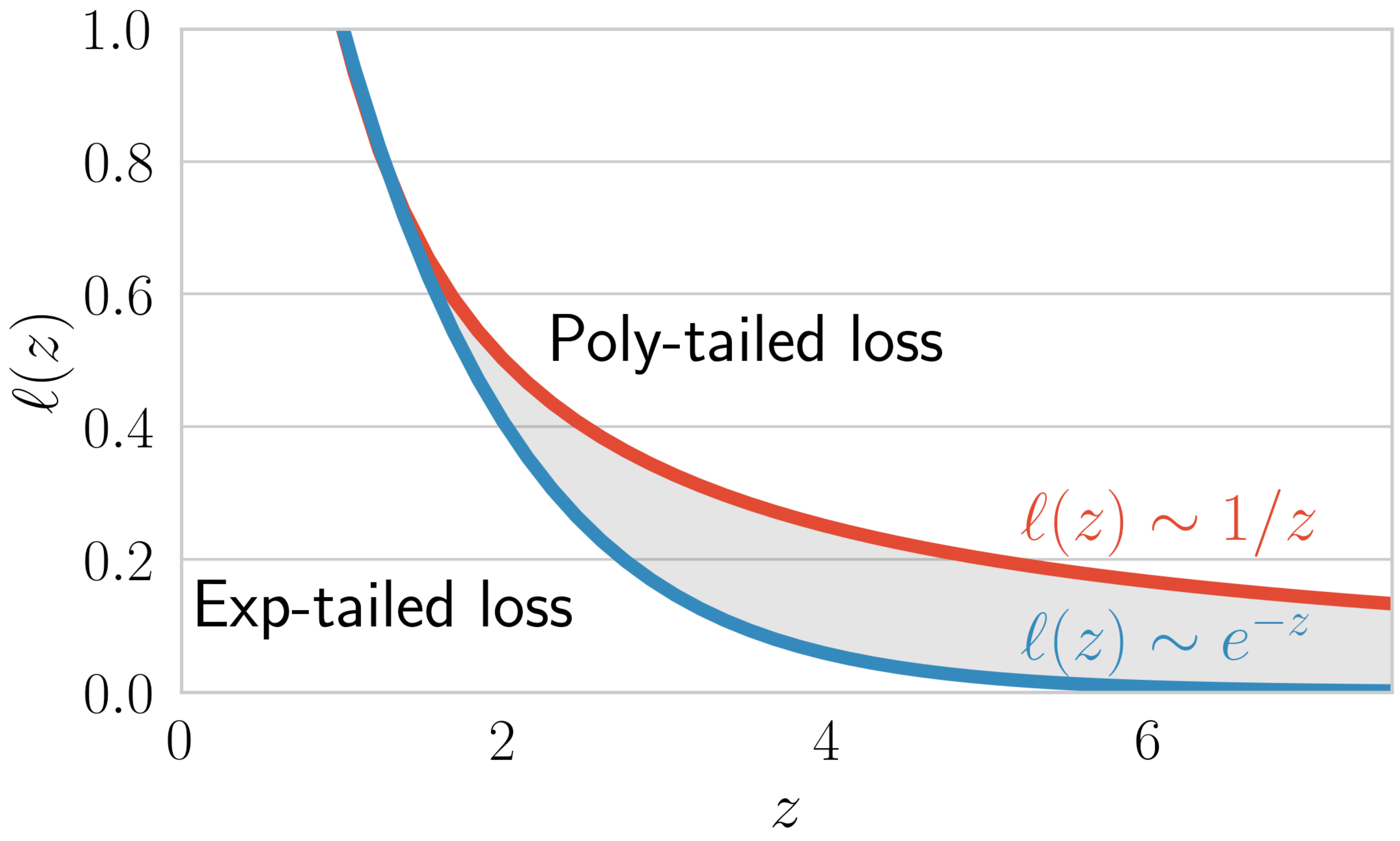
$$\underbrace{(x_i, y_i), \ldots, (x_i, y_i)}_{w_i \text{ times}}$$

The max-margin classifier for this new dataset is *unchanged*

$$\arg\max_{\|\theta\|_2 = 1} \left\{\gamma \ : \ \text{subject to } y_i x_i^\top \theta \geq \gamma, \forall i \in [n]\right\}$$

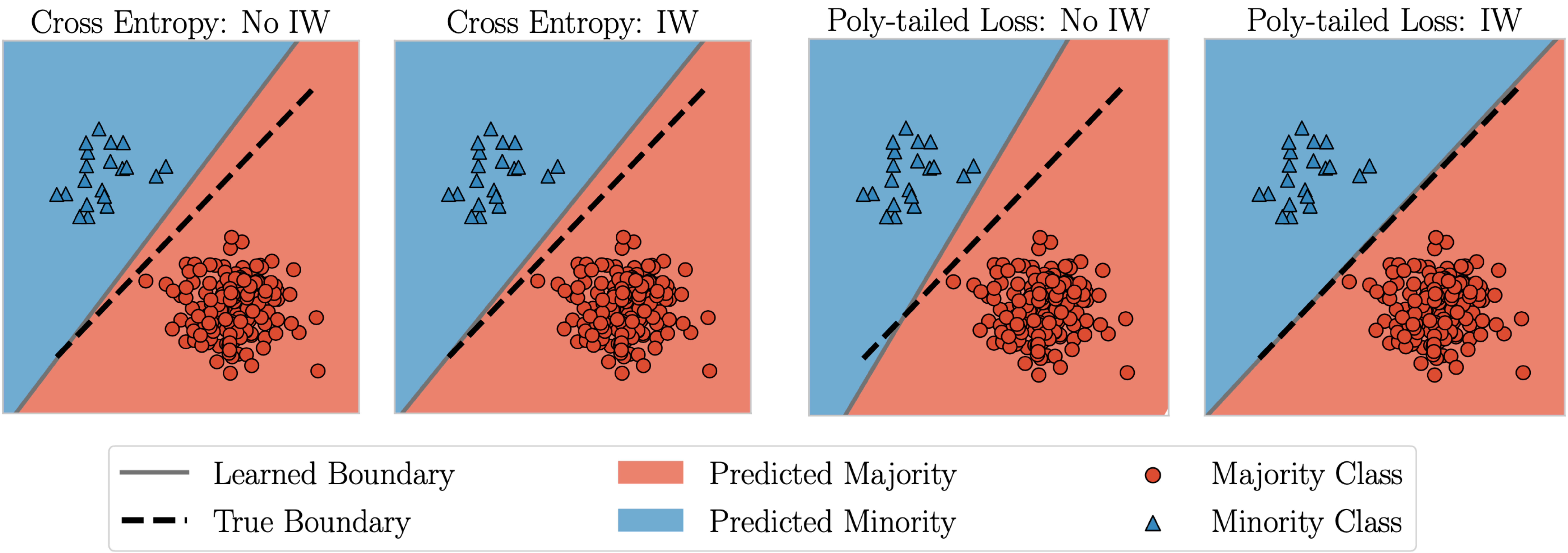
Prior implicit bias results implies $t \to \infty$ reweighting is ineffective (Soudry et al. 2018, Ji and Telgarsky 2018)

Our Proposal: Switch losses $\log(1+\exp(-yf_\theta(x))) \longrightarrow \frac{1}{yf_\theta(x)}$

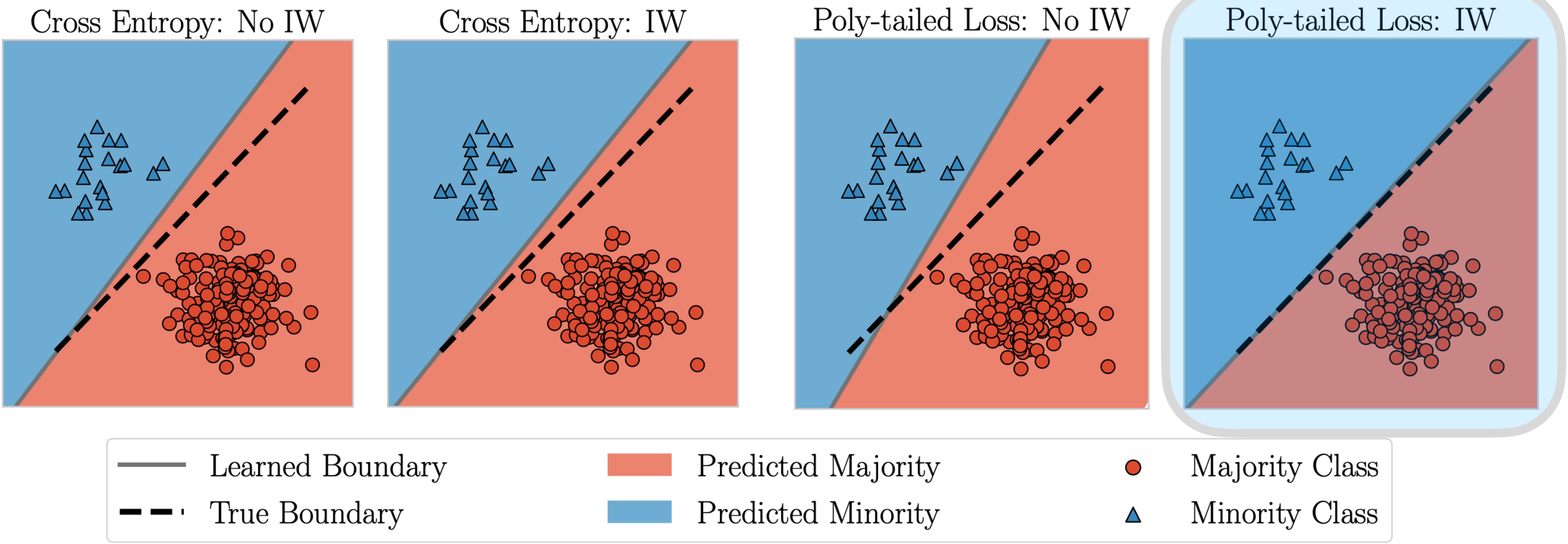


Our Proposal: Switch losses

$$\log(1 + \exp(-yf_\theta(x))) \longrightarrow \frac{1}{yf_\theta(x)}$$



Our Proposal: Switch losses $\log(1+\exp(-yf_\theta(x))) \longrightarrow \frac{1}{yf_\theta(x)}$



We provably show it has the correct implicit bias

Implicit bias for poly-tailed losses

Implicit bias for poly-tailed losses

If gradient descent used to minimize linearly separable $(x_1, y_1), \ldots, (x_n, y_n)$

$$L(\theta) = \sum_{i=1}^{n} w_i \ell(y_i x_i^\top \theta) \qquad \ell(z) = \begin{cases} \dfrac{\log(1+\exp(-z))}{\log(1+\exp(-1))} & z \le 1 \\ \boxed{\dfrac{1}{z^\alpha}} & z > 1 \end{cases}$$

Implicit bias for poly-tailed losses

If gradient descent used to minimize linearly separable $(x_1, y_1), \ldots, (x_n, y_n)$

$$L(\theta) = \sum_{i=1}^{n} w_i \ell(y_i x_i^\top \theta) \qquad \ell(z) = \begin{cases} \dfrac{\log(1+\exp(-z))}{\log(1+\exp(-1))} & z \leq 1 \\ \dfrac{1}{z^\alpha} & z > 1 \end{cases}$$

Given iterates $\theta^{(t+1)} = \theta^{(t)} - \eta \nabla L(\theta^{(t)})$ if η is small enough

Implicit bias for poly-tailed losses

If gradient descent used to minimize linearly separable $(x_1, y_1), \ldots, (x_n, y_n)$

$$L(\theta) = \sum_{i=1}^{n} w_i \ell(y_i x_i^\top \theta) \qquad \ell(z) = \begin{cases} \frac{\log(1+\exp(-z))}{\log(1+\exp(-1))} & z \le 1 \\ \frac{1}{z^\alpha} & z > 1 \end{cases}$$

Given iterates $\theta^{(t+1)} = \theta^{(t)} - \eta \nabla L(\theta^{(t)})$ if η is small enough

$$\frac{\theta^{(t)}}{\|\theta^{(t)}\|} \to \arg \min_{\|\theta\|_2 = 1} \left\{ \sum_{i \in [n]} \frac{w_i}{(y_i x_i^\top \theta)^\alpha} \ : \ \text{subject to } y_i x_i^\top \theta > 0 \right\}$$

Implicit bias for poly-tailed losses

If gradient descent used to minimize linearly separable $(x_1, y_1), \ldots, (x_n, y_n)$

$$L(\theta) = \sum_{i=1}^{n} w_i \ell(y_i x_i^\top \theta) \qquad \ell(z) = \begin{cases} \dfrac{\log(1+\exp(-z))}{\log(1+\exp(-1))} & z \leq 1 \\ \boxed{\dfrac{1}{z^\alpha}} & z > 1 \end{cases}$$

Given iterates $\theta^{(t+1)} = \theta^{(t)} - \eta \nabla L(\theta^{(t)})$ if η is small enough

$$\frac{\theta^{(t)}}{\|\theta^{(t)}\|} \to \boxed{\arg\min_{\|\theta\|_2 = 1} \left\{ \sum_{i \in [n]} \frac{w_i}{(y_i x_i^\top \theta)^\alpha} \ : \ \text{subject to } y_i x_i^\top \theta > 0 \right\}}$$

"poly-tailed classifier" θ_α

Implicit bias for poly-tailed losses

If gradient descent used to minimize linearly separable $(x_1, y_1), \ldots, (x_n, y_n)$

$$L(\theta) = \sum_{i=1}^{n} w_i \ell(y_i x_i^\top \theta) \qquad \ell(z) = \begin{cases} \dfrac{\log(1+\exp(-z))}{\log(1+\exp(-1))} & z \le 1 \\ \boxed{\dfrac{1}{z^\alpha}} & z > 1 \end{cases}$$

Given iterates $\theta^{(t+1)} = \theta^{(t)} - \eta \nabla L(\theta^{(t)})$ if η is small enough

$$\frac{\theta^{(t)}}{\|\theta^{(t)}\|} \to \arg\min_{\|\theta\|_2=1} \left\{ \sum_{i\in[n]} \frac{\boxed{w_i}}{(y_i x_i^\top \theta)^\alpha} \; : \; \text{subject to } y_i x_i^\top \theta > 0 \right\}$$

"poly-tailed classifier" θ_α

Maximizes a sum of **weighted margins**

Implicit bias for poly-tailed losses

If gradient descent used to minimize linearly separable $(x_1, y_1), \ldots, (x_n, y_n)$

$$L(\theta) = \sum_{i=1}^{n} w_i \ell(y_i x_i^\top \theta) \qquad \ell(z) = \begin{cases} \dfrac{\log(1+\exp(-z))}{\log(1+\exp(-1))} & z \leq 1 \\ \boxed{\dfrac{1}{z^\alpha}} & z > 1 \end{cases}$$

Given iterates $\theta^{(t+1)} = \theta^{(t)} - \eta \nabla L(\theta^{(t)})$ if η is small enough

$$\frac{\theta^{(t)}}{\|\theta^{(t)}\|} \to \boxed{\arg\min_{\|\theta\|_2=1} \left\{ \sum_{i\in[n]} \frac{\boxed{w_i}}{(y_i x_i^\top \theta)^\alpha} \ : \ \text{subject to } y_i x_i^\top \theta > 0 \right\}}$$

"poly-tailed classifier" θ_α

Builds on results by Ji et al. 2020

Maximizes a sum of **weighted margins**

But what about the test performance?

But what about the test performance?

Does maximizing the weighted margin translate into robust test accuracy?

But what about the test performance?

Does maximizing the weighted margin translate into robust test accuracy?

What's coming up…

But what about the test performance?

Does maximizing the weighted margin translate into robust test accuracy?

What's coming up…

1. Setting where the poly-tailed classifier achieves minimax accuracy

But what about the test performance?

Does maximizing the weighted margin translate into robust test accuracy?

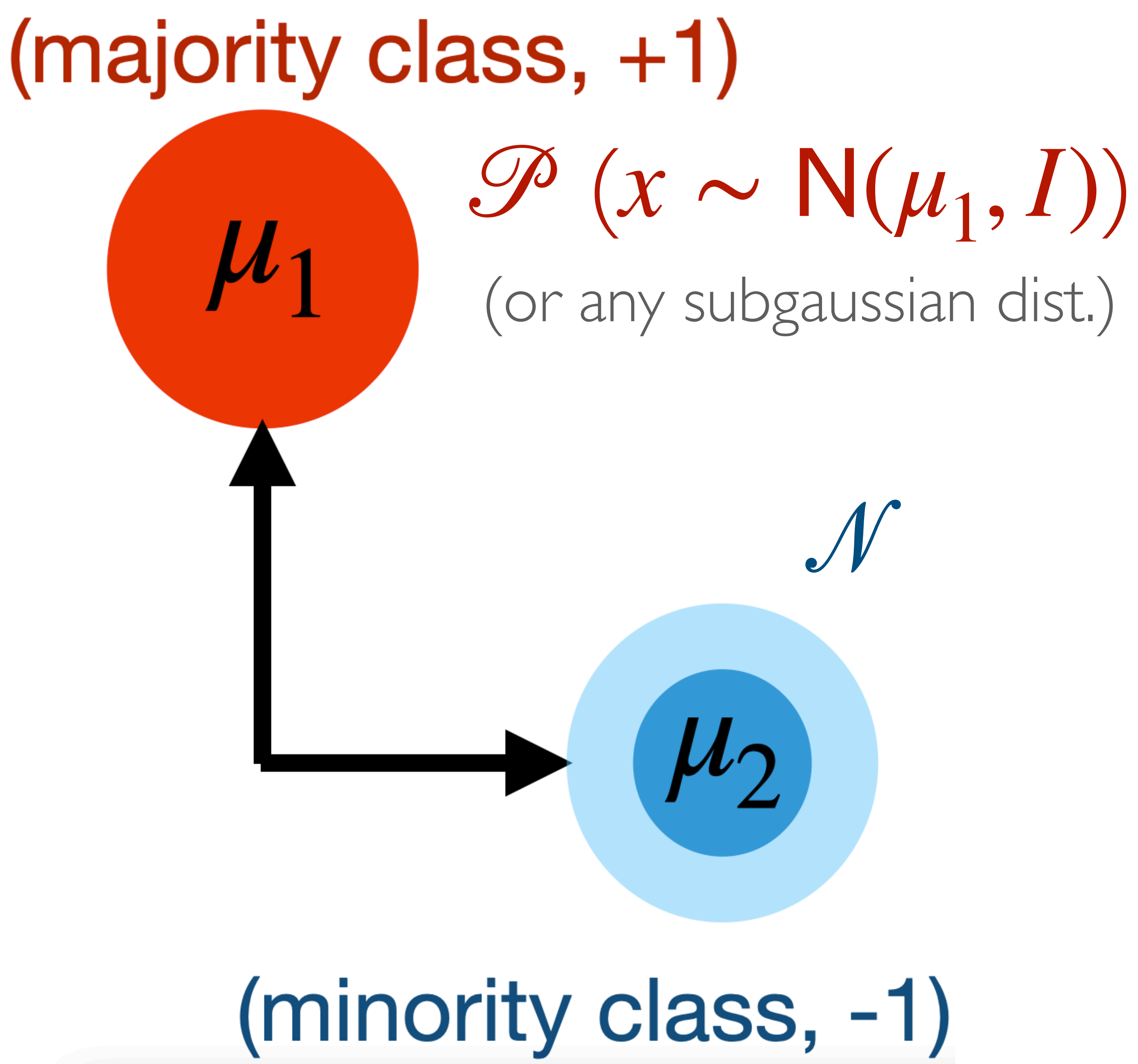
What's coming up…

1. Setting where the poly-tailed classifier achieves minimax accuracy

2. A lower bound that shows that the max-margin classifier fails

What about the Test Error?

Want to study the generalization error in the overparameterized regime with distribution shift

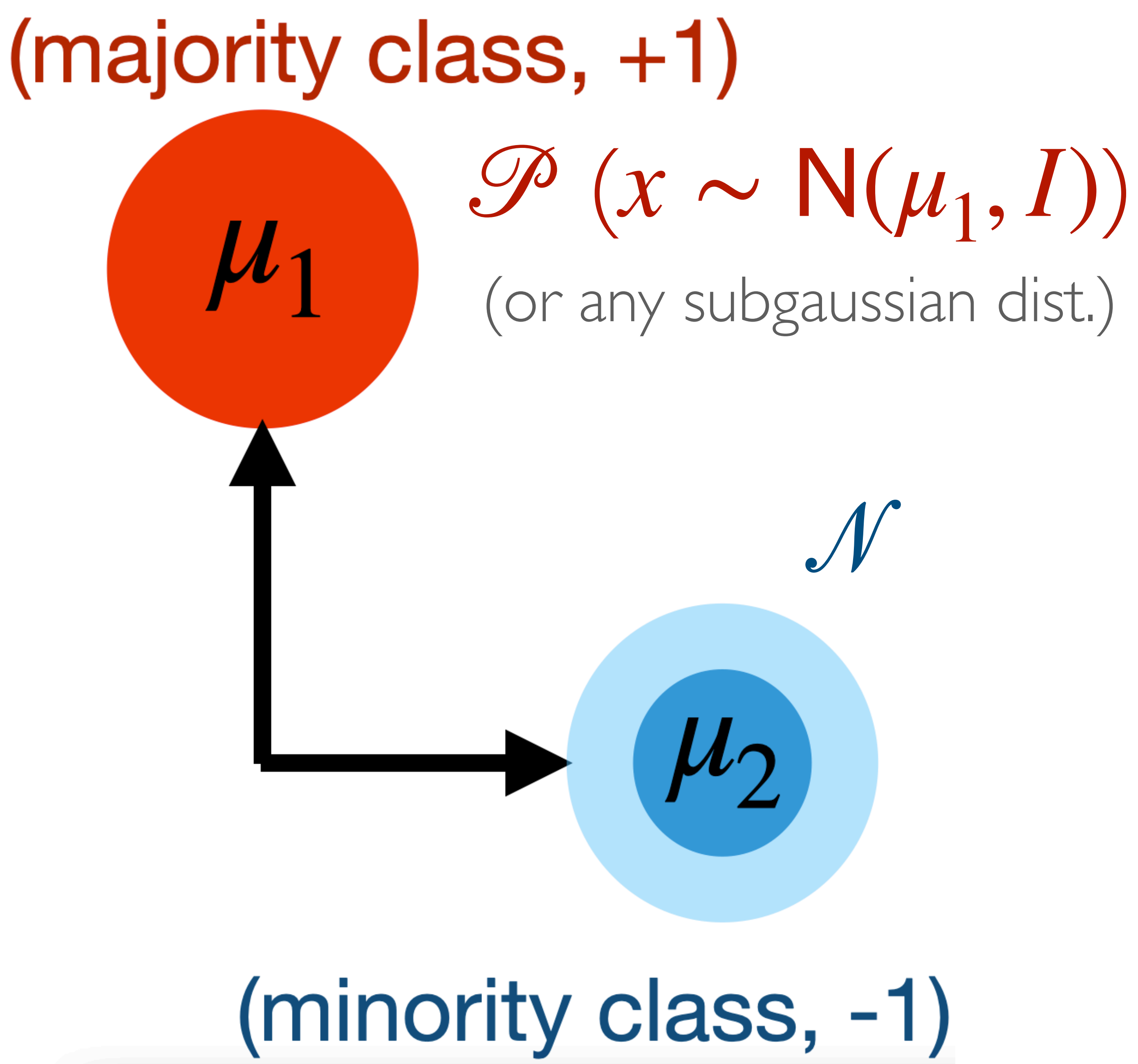


(C. & Long 2020, Cao et al. 2021)

What about the Test Error?

Want to study the generalization error in the overparameterized regime with distribution shift

Skewed data with $|\mathscr{P}| \geq |\mathscr{N}|$, with $\tau = \dfrac{|\mathscr{P}|}{|\mathscr{N}|}$



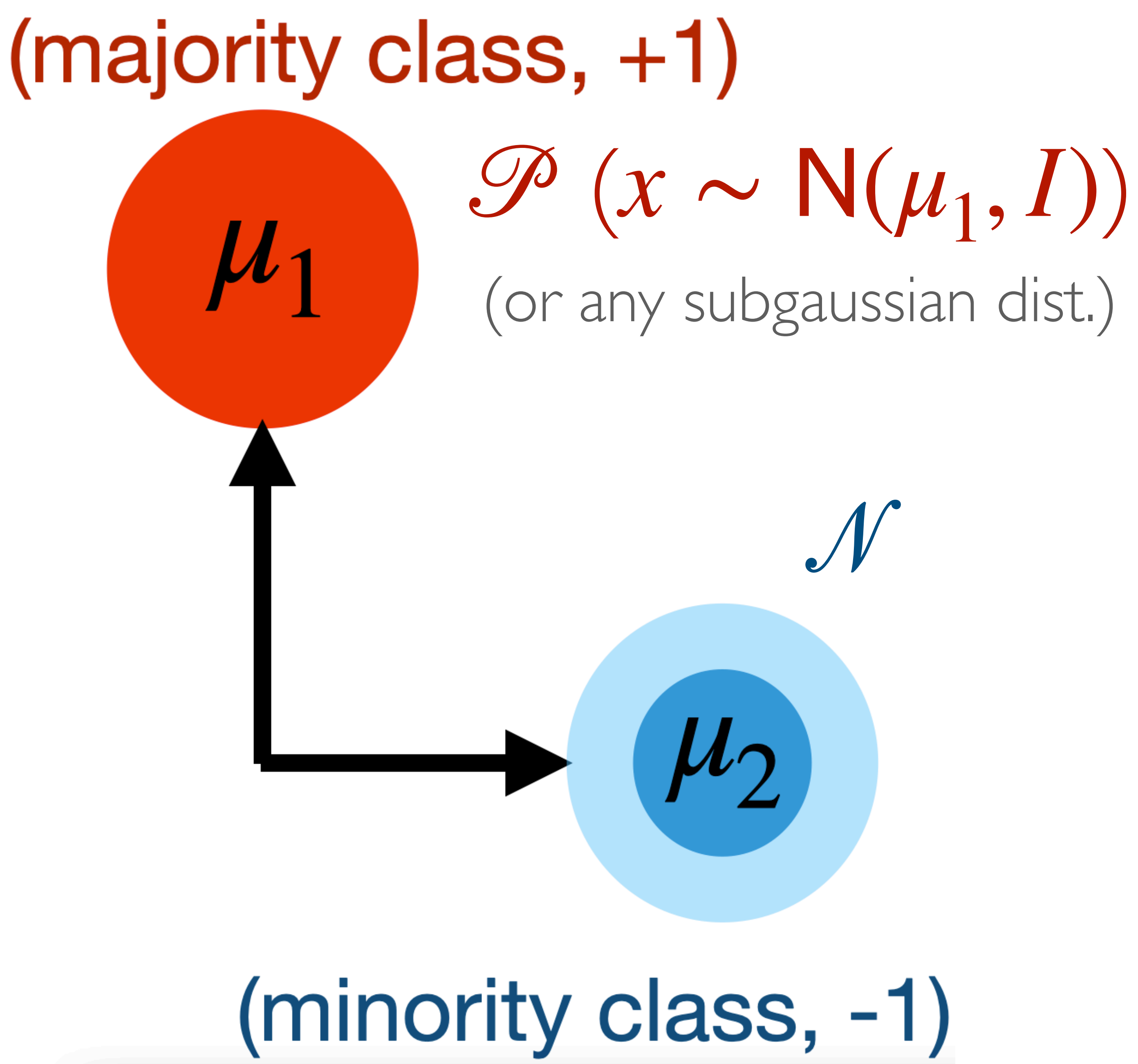
(C. & Long 2020, Cao et al. 2021)

What about the Test Error?

Want to study the generalization error in the overparameterized regime with distribution shift

Skewed data with $|\mathscr{P}| \geq |\mathscr{N}|$, with $\tau = \dfrac{|\mathscr{P}|}{|\mathscr{N}|}$

Test data is uniform mixture



(C. & Long 2020, Cao et al. 2021)

What about the Test Error?

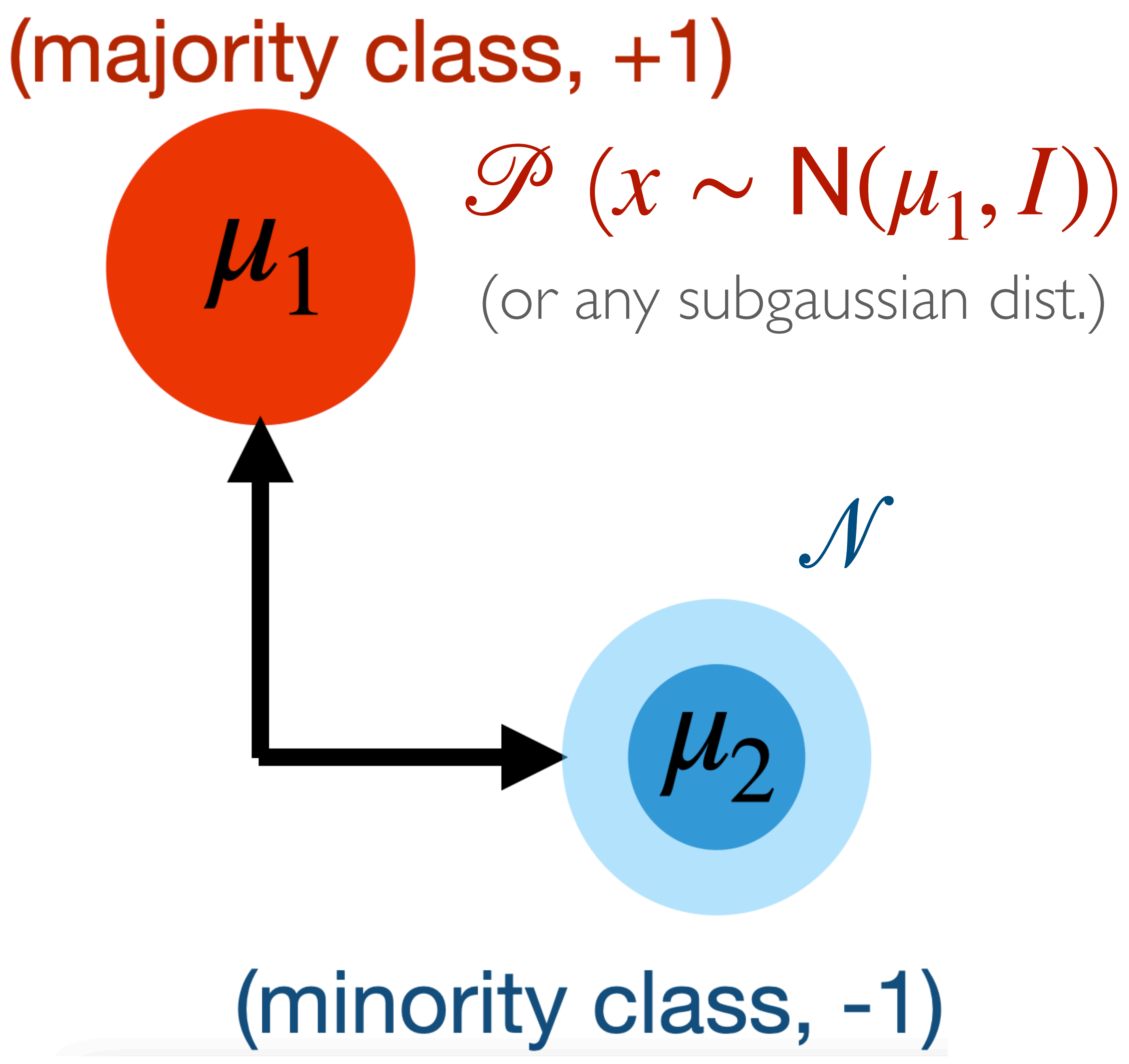
Want to study the generalization error in the overparameterized regime with distribution shift

Skewed data with $|\mathscr{P}| \geq |\mathscr{N}|$, with $\tau = \dfrac{|\mathscr{P}|}{|\mathscr{N}|}$

Test data is uniform mixture

Assumptions on the data

- $n \geq C \log(1/\delta)$
- $\|\mu\|^2 \geq Cn^2 \log(n/\delta)$
- $d \geq Cn\|\mu\|^2$ (high dim. setting)



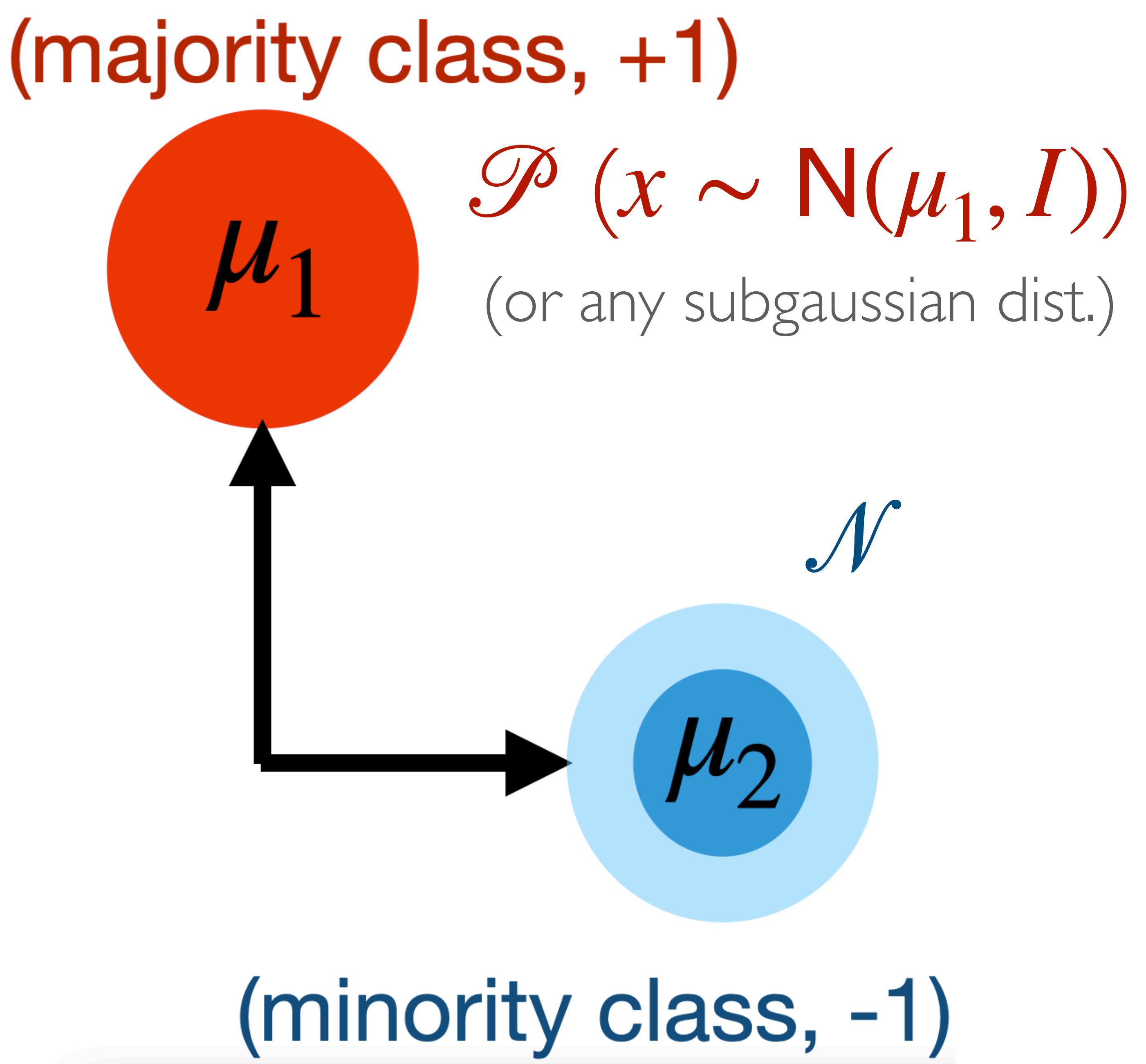
(C. & Long 2020, Cao et al. 2021)

What about the Test Error?

Want to study the generalization error in the overparameterized regime with distribution shift

Skewed data with $|\mathscr{P}| \geq |\mathscr{N}|$, with $\tau = \dfrac{|\mathscr{P}|}{|\mathscr{N}|}$

Test data is uniform mixture



(C. & Long 2020, Cao et al. 2021)

Assumptions on the data

- $n \geq C \log(1/\delta)$
- $\|\mu\|^2 \geq Cn^2 \log(n/\delta)$
- $d \geq Cn\|\mu\|^2$ (high dim. setting)

Set weights as $w_i = \begin{cases} 1 & \text{if } i \in \mathscr{P} \\ w > 1 & \text{if } i \in \mathscr{N} \end{cases}$

Test Error of the Poly-tailed Classifiers

$$\ell(z) \sim \frac{1}{z}$$

Test Error of the Poly-tailed Classifiers

$$\ell(z) \sim \frac{1}{z}$$

Theorem: There exists a constant c such that for all large enough C, for any $\delta < 1/C$, if the weight

Test Error of the Poly-tailed Classifiers

$$\ell(z) \sim \frac{1}{z}$$

Theorem: There exists a constant c such that for all large enough C, for any $\delta < 1/C$, if the weight $\boxed{\frac{\tau^3}{2} \leq w \leq 2\tau^3}$

Test Error of the Poly-tailed Classifiers

$$\ell(z) \sim \frac{1}{z}$$

Theorem: There exists a constant c such that for all large enough C, for any $\delta < 1/C$, if the weight

$$\frac{\tau^3}{2} \leq w \leq 2\tau^3$$

then with probability at least $1 - \delta$

$$\text{TestError}(\theta_1) \leq \exp\left(-\frac{c\,|\mathcal{N}|\,\|\mu\|^4}{d}\right).$$

Test Error of the Poly-tailed Classifiers

$$\ell(z) \sim \frac{1}{z}$$

Theorem: There exists a constant c such that for all large enough C, for any $\delta < 1/C$, if the weight

$$\frac{\tau^3}{2} \le w \le 2\tau^3$$

then with probability at least $1 - \delta$

$$\text{TestError}(\theta_1) \le \exp\left(-\frac{c\,|\mathcal{N}|\,\|\mu\|^4}{d}\right). \to 0 \text{ if } \frac{\sqrt{|\mathcal{N}|}\,\|\mu\|^2}{\sqrt{d}} \to \infty$$

(minimax optimal, Giraud and Verzelen 2019)

Test Error of the Poly-tailed Classifiers

$$\ell(z) \sim \frac{1}{z}$$

Theorem: There exists a constant c such that for all large enough C, for any $\delta < 1/C$, if the weight

$$\frac{\tau^3}{2} \leq w \leq 2\tau^3$$

then with probability at least $1 - \delta$

$$\text{TestError}(\theta_1) \leq \exp\left(-\frac{c\,|\mathcal{N}|\,\|\mu\|^4}{d}\right). \to 0 \text{ if } \frac{\sqrt{|\mathcal{N}|}\,\|\mu\|^2}{\sqrt{d}} \to \infty$$

(minimax optimal, Giraud and Verzelen 2019)

Further, if the imbalance τ is sufficiently large then w.p. at least $1 - \delta$

$$\text{TestError}(\theta_{\text{MM}}) \geq \frac{1}{8}.$$

Separation between poly-tailed and exp-tailed classifiers

Example setting:

- $||\mu||^2 = d^{\frac{1}{2}+\frac{1}{40}}$
- $|\mathcal{N}| = d^{\frac{1}{5}}$
- $\tau = d^{\frac{3}{20}}$

Separation between poly-tailed and exp-tailed classifiers

Example setting:
- $\|\mu\|^2 = d^{\frac{1}{2}+\frac{1}{40}}$
- $|\mathcal{N}| = d^{\frac{1}{5}}$
- $\tau = d^{\frac{3}{20}}$

As $d \to \infty$ (w.h.p.)

$$\text{TestError}(\theta_{\text{MM}}) \geq \frac{1}{8} \geq \text{TestError}(\theta_1) \to 0$$

(IW exp-tailed classifier) (IW poly-tailed classifier)

Separation between poly-tailed and exp-tailed classifiers

Example setting:

- $\|\mu\|^2 = d^{\frac{1}{2}+\frac{1}{40}}$
- $|\mathcal{N}| = d^{\frac{1}{5}}$
- $\tau = d^{\frac{3}{20}}$

As $d \to \infty$ (w.h.p.)

$$\text{TestError}(\theta_{\text{MM}}) \geq \frac{1}{8} \geq \text{TestError}(\theta_1) \to 0$$

(IW exp-tailed classifier) (IW poly-tailed classifier)

Importance weighted poly-tailed classifier provably generalizes better

Why this choice of weight w?

Theorem: There exists a constant c such that for all large enough C, for any $\delta < 1/C$, if the weight

$$\frac{\tau^3}{2} \leq w \leq 2\tau^3$$

then with probability at least $1 - \delta$

$$\text{TestError}(\theta_1) \leq \exp\left(-\frac{c\,|\mathcal{N}|\,\|\mu\|^4}{d}\right).$$

Further, if the imbalance τ is sufficiently large then w.p. at least $1 - \delta$

$$\text{TestError}(\theta_{\text{MM}}) \geq \frac{1}{8}.$$

Why this choice of weight w?

$$\frac{\tau^3}{2} \le w \le 2\tau^3$$

Why this choice of weight w?

$$\frac{\tau^3}{2} \leq w \leq 2\tau^3$$

This choice is unusual since the resulting loss is *biased*

Why this choice of weight w?

$$\frac{\tau^3}{2} \leq w \leq 2\tau^3$$

This choice is unusual since the resulting loss is *biased*

Classical choice $w = \tau$ leads to unbiased training loss

Why this choice of weight w?

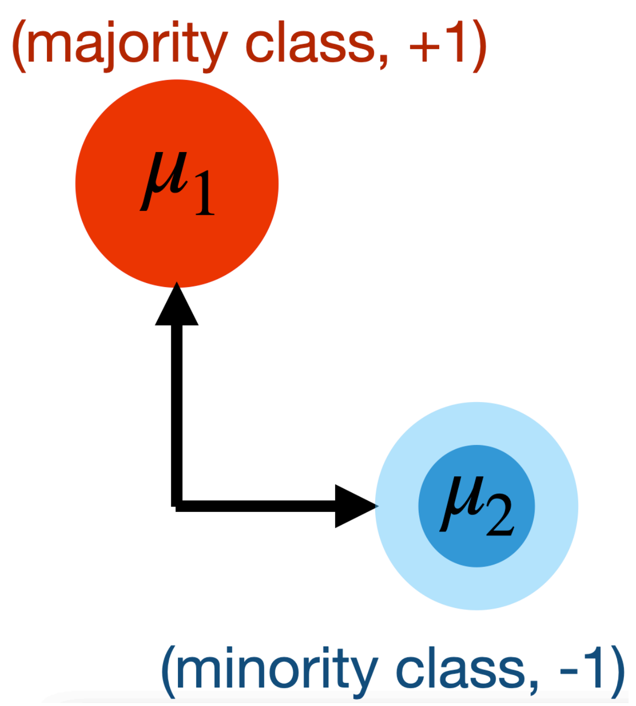
$$\frac{\tau^3}{2} \leq w \leq 2\tau^3$$

This choice is unusual since the resulting loss is *biased*

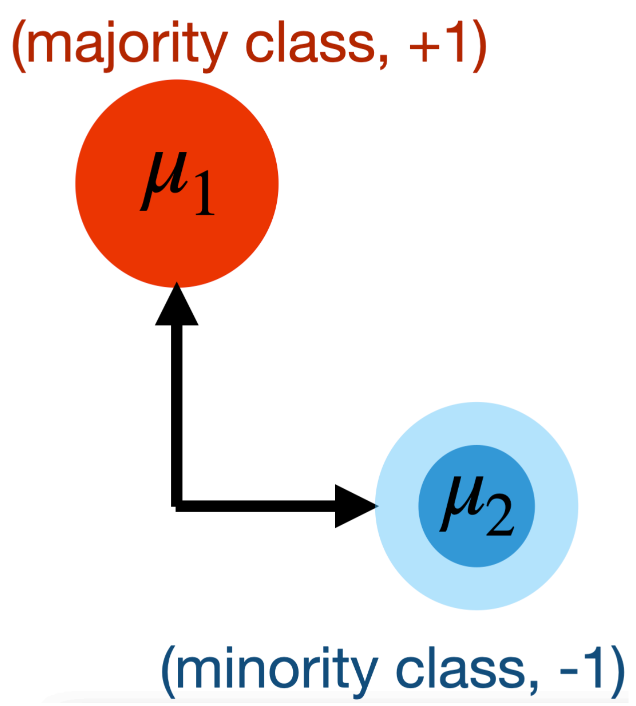
Classical choice $w = \tau$ leads to unbiased training loss

Nothing special about τ^3, if $L(z) \sim \dfrac{1}{z^\alpha}$, then $w \asymp \tau^{\frac{\alpha(\alpha+2)}{\alpha^2+\alpha-1}}$

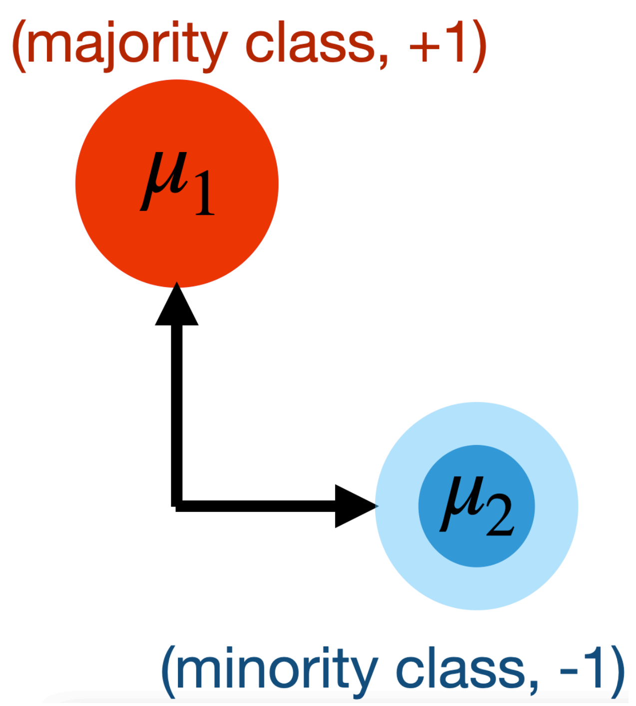
Exponentiate the weights and train on biased loss

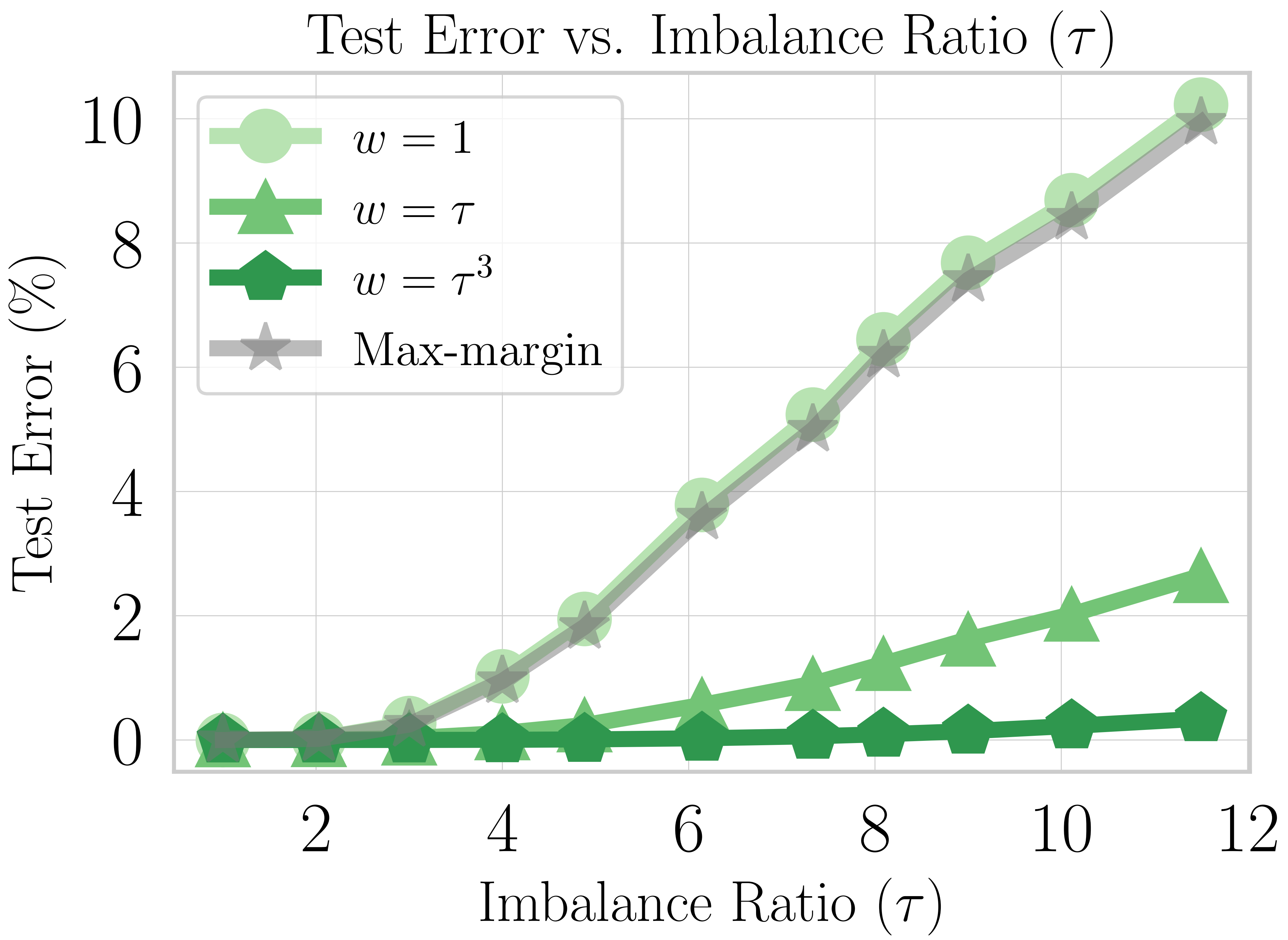


Exponentiate the weights and train on biased loss



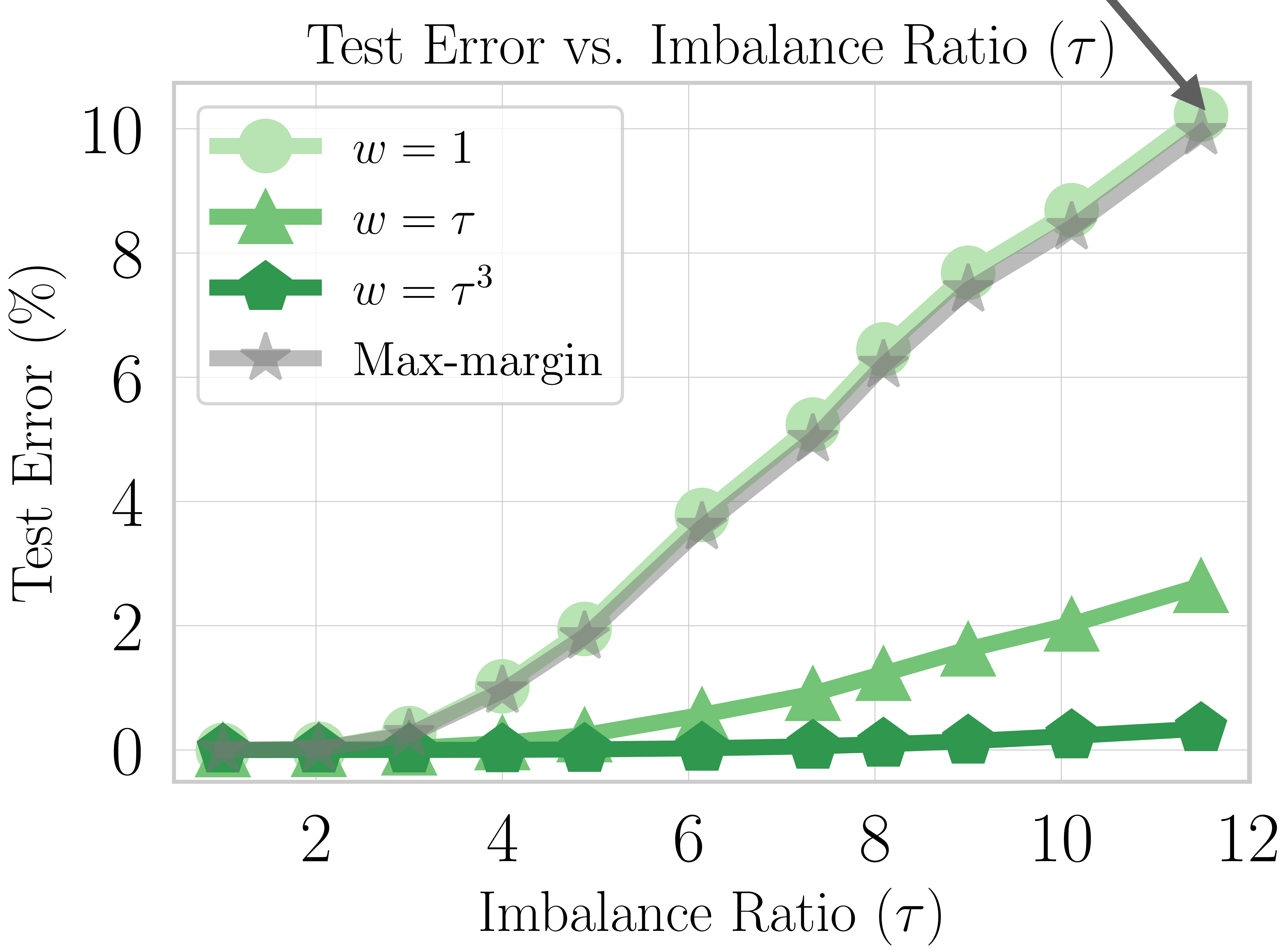
Exponentiate the weights and train on biased loss

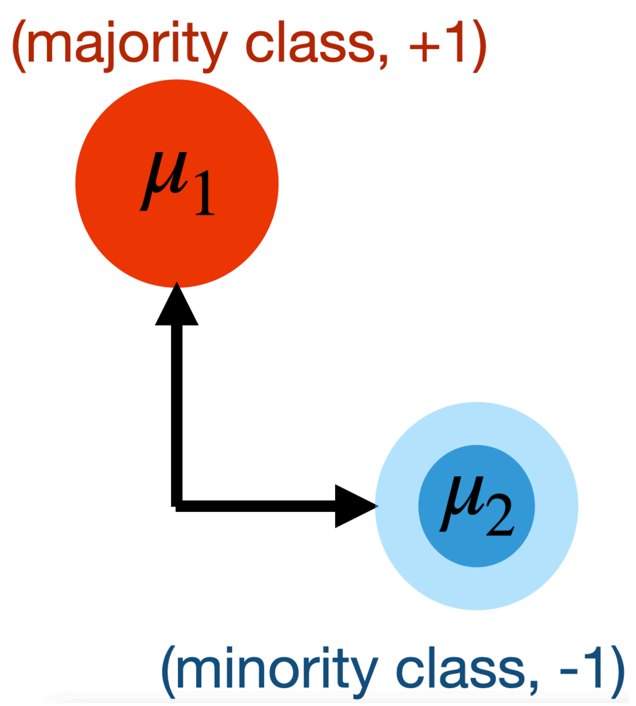




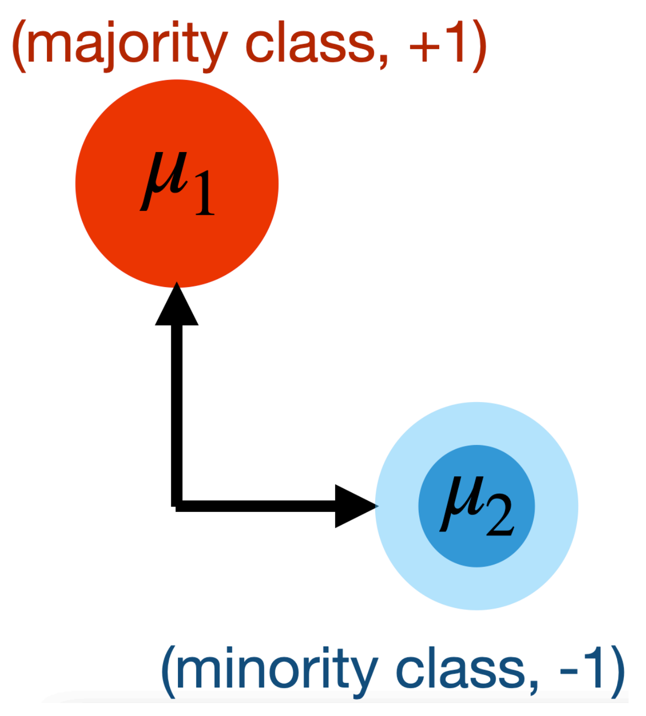
Exponentiate the weights and train on biased loss

max. margin & no IW

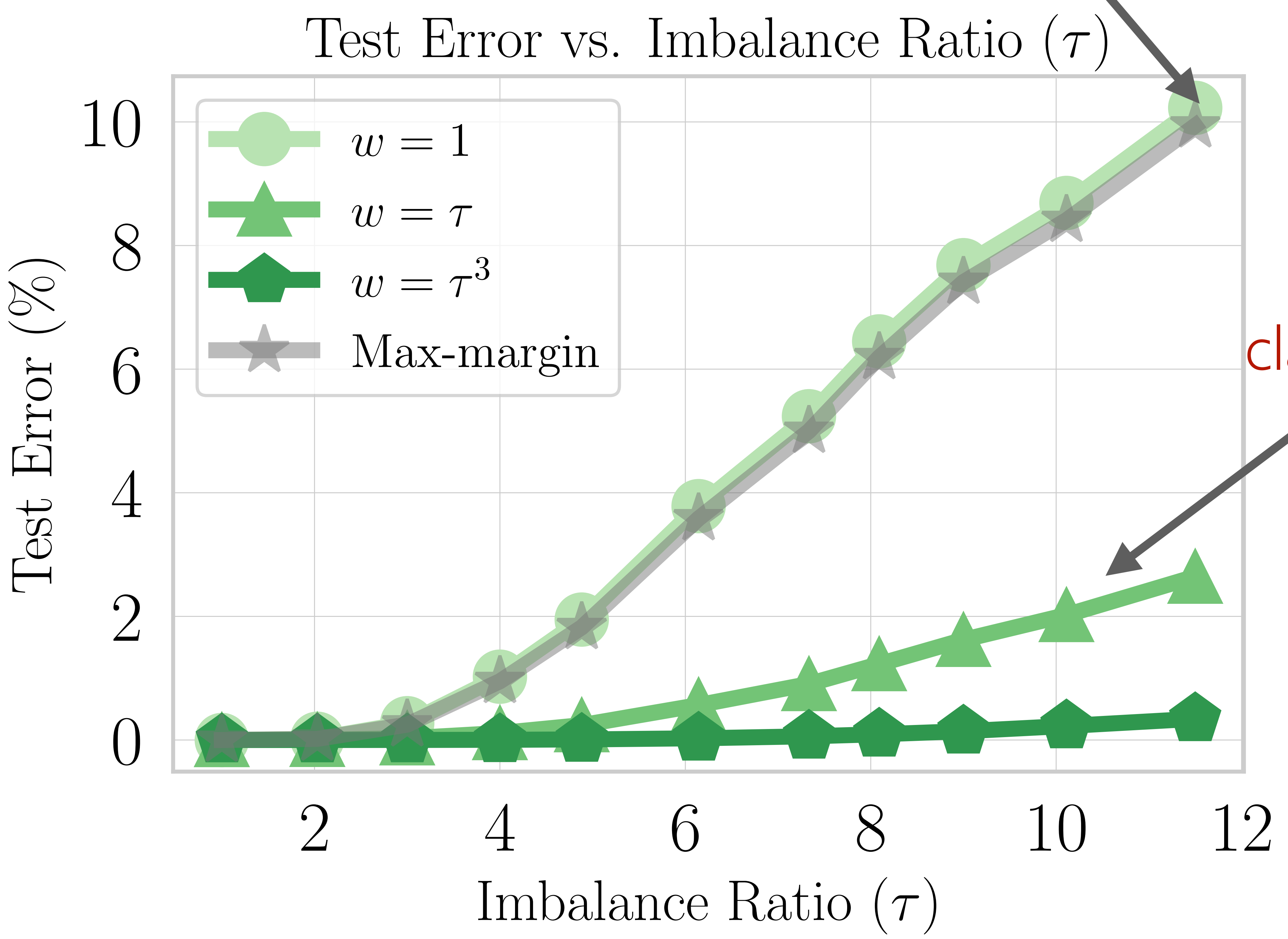




Exponentiate the weights and train on biased loss

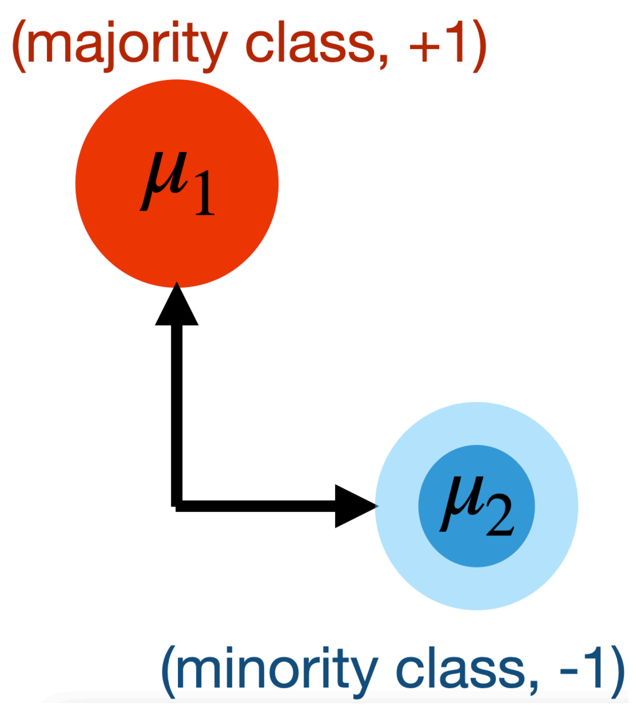


max. margin & no IW

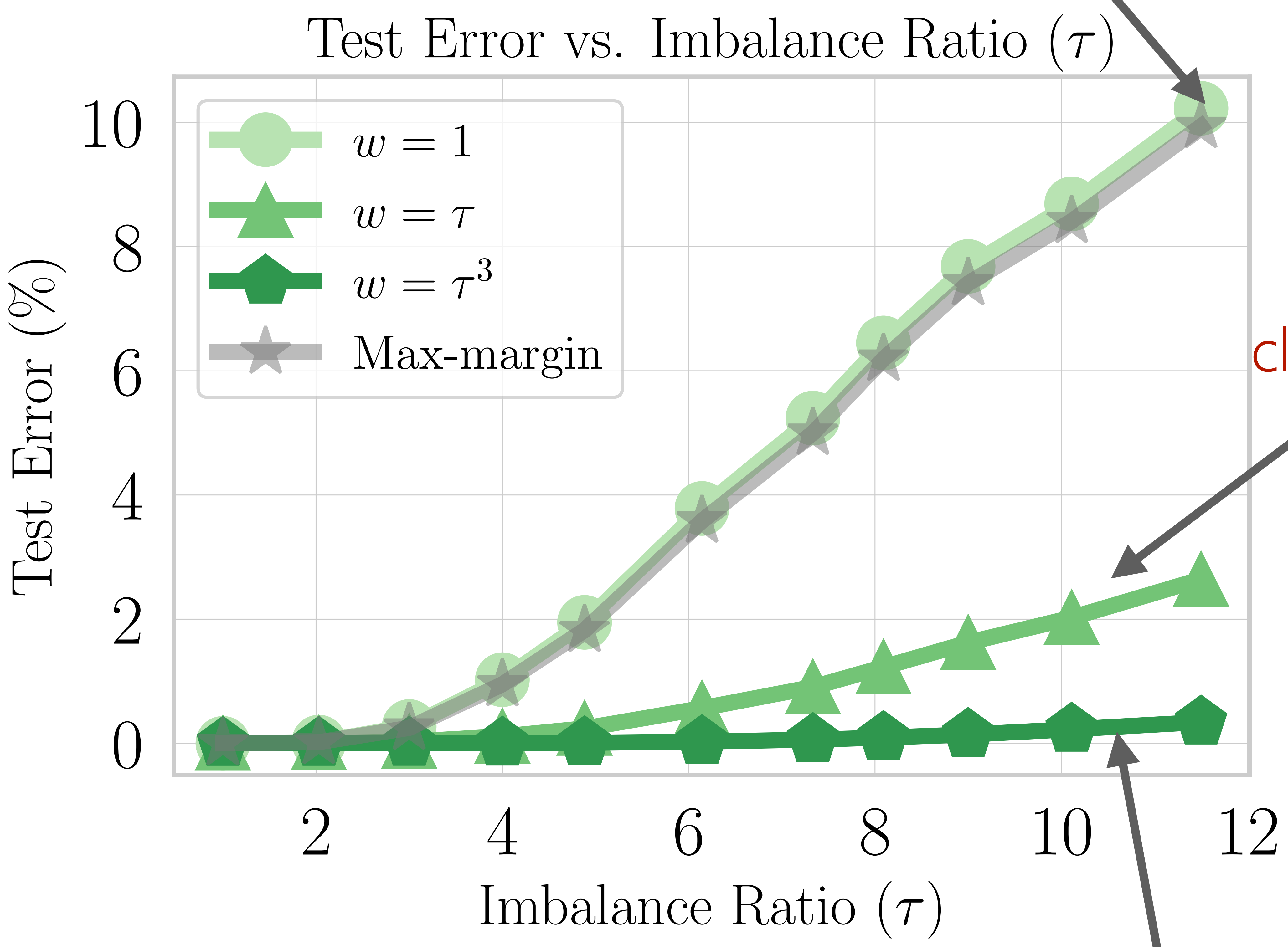


classic IW
$w = \tau$

Exponentiate the weights and train on biased loss



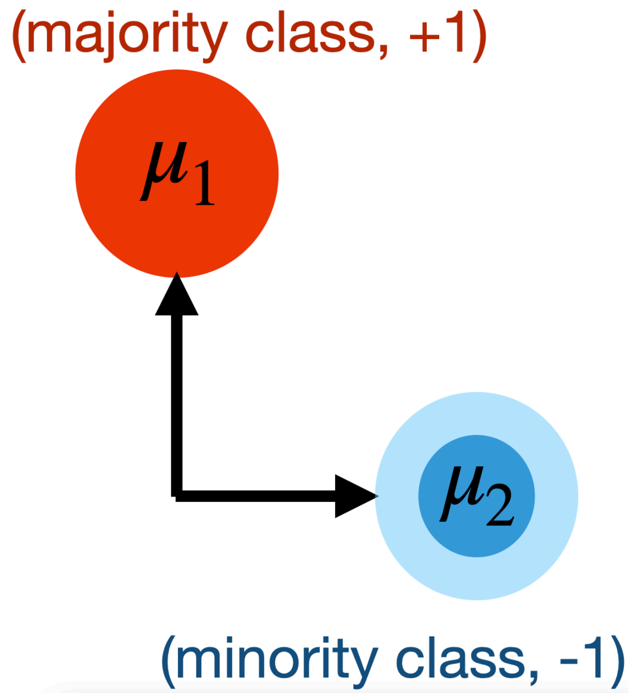
max. margin & no IW



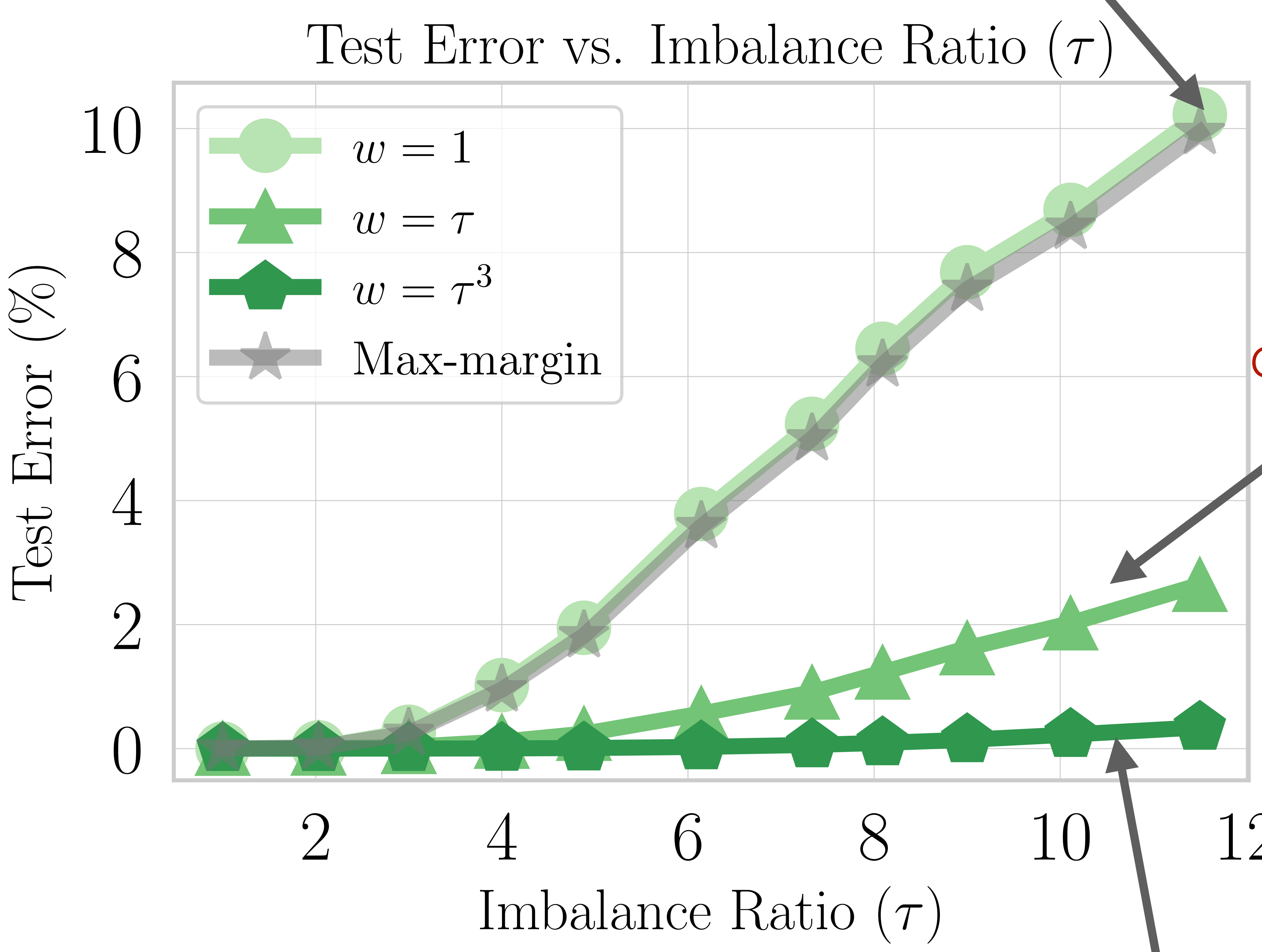
classic IW
$w = \tau$

low error when $w = \tau^3$

Exponentiate the weights and train on biased loss

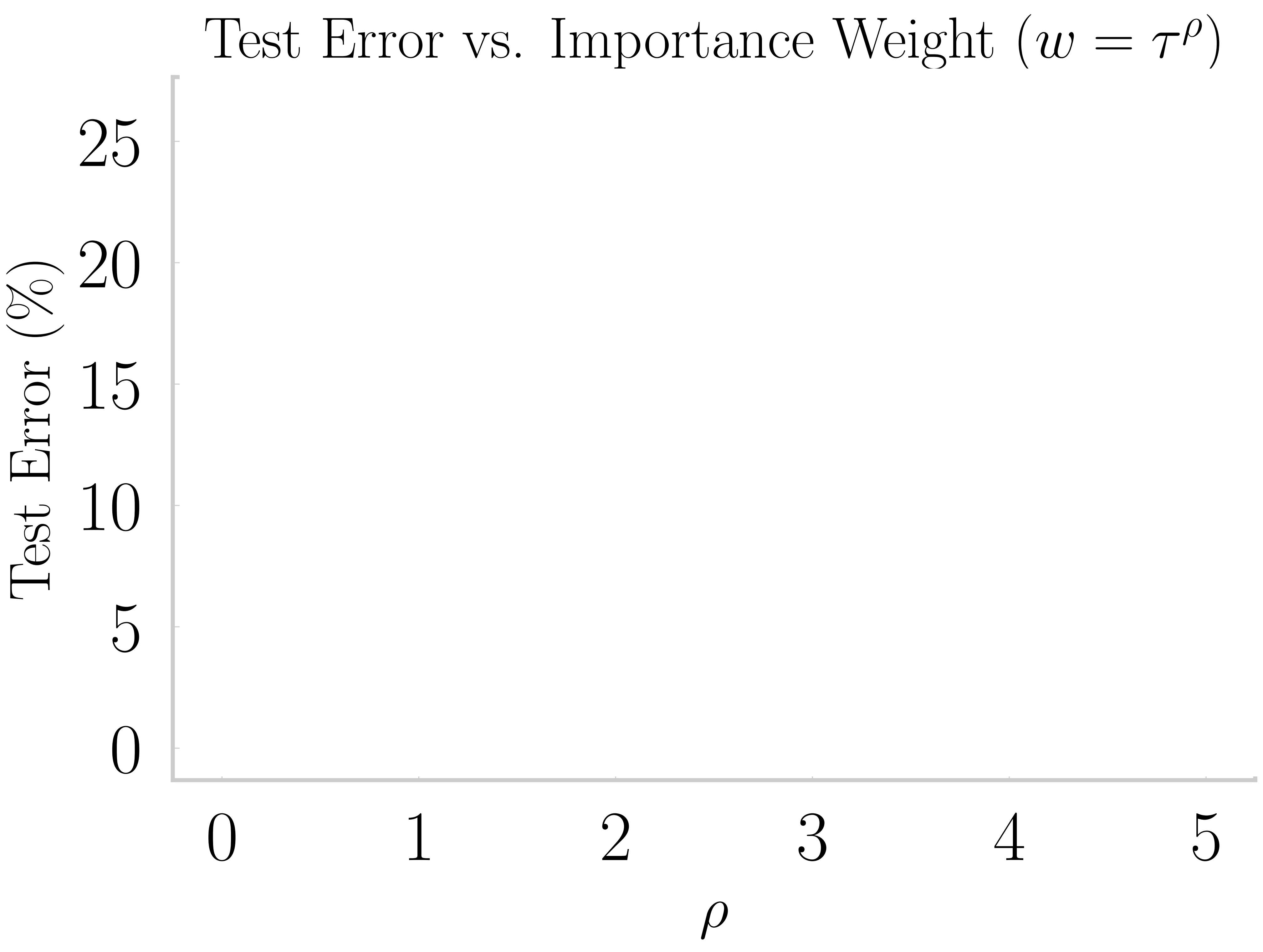


max. margin & no IW

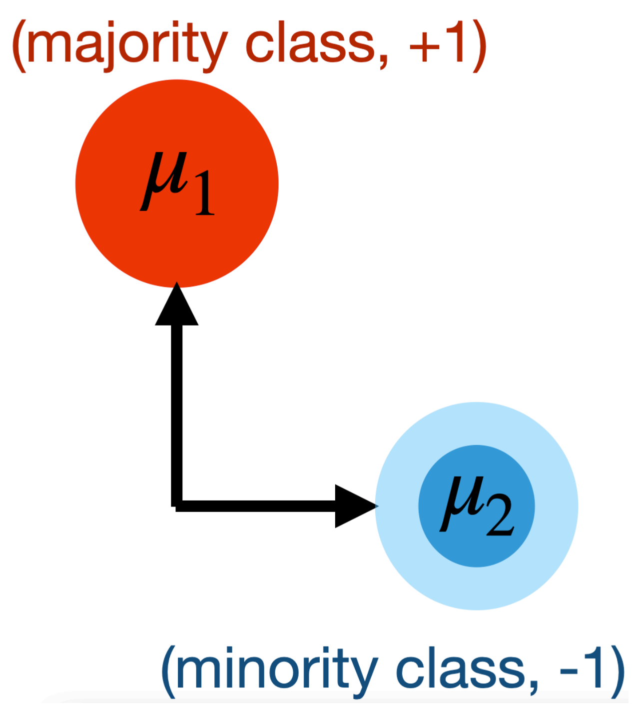


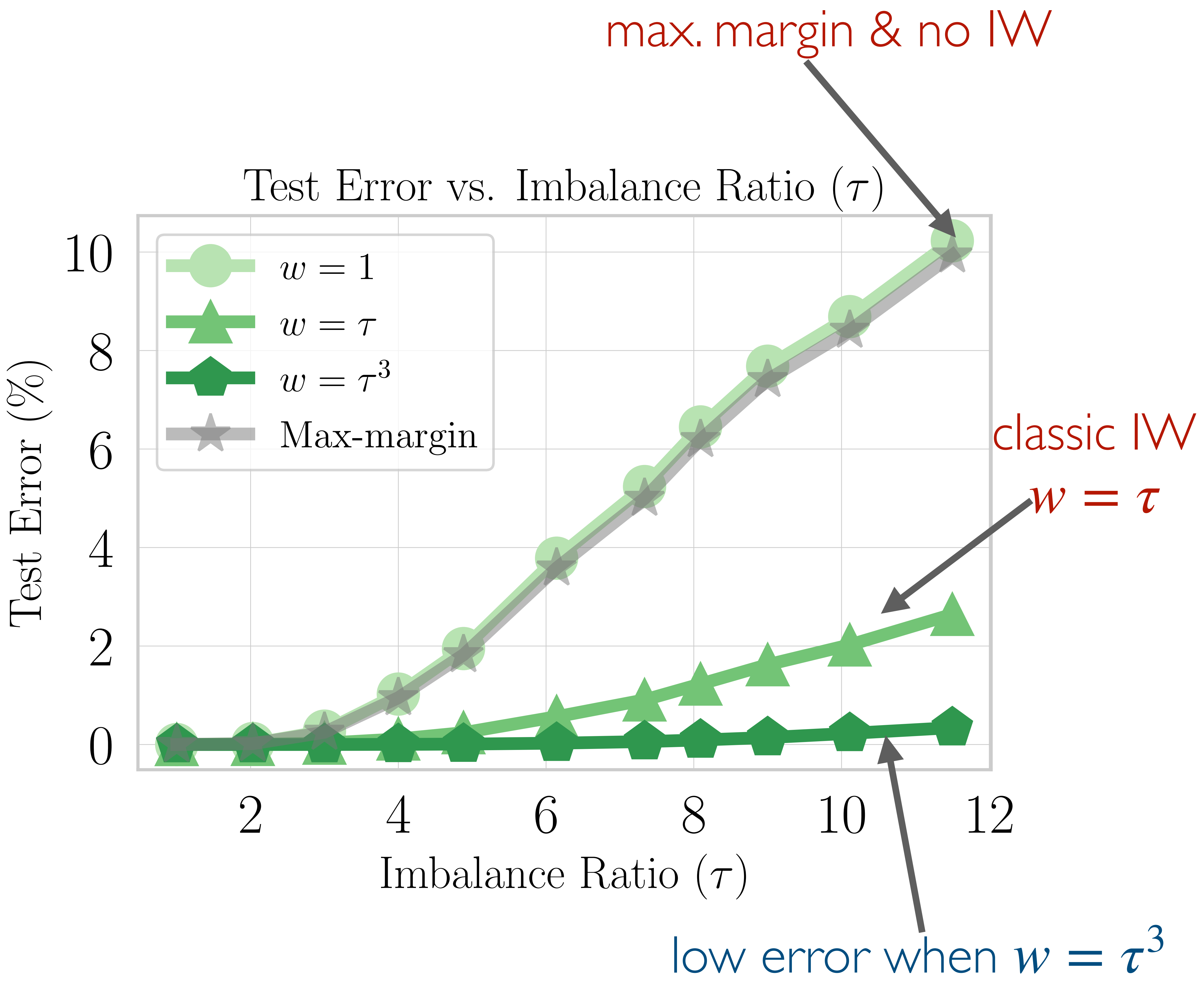
classic IW
$w = \tau$

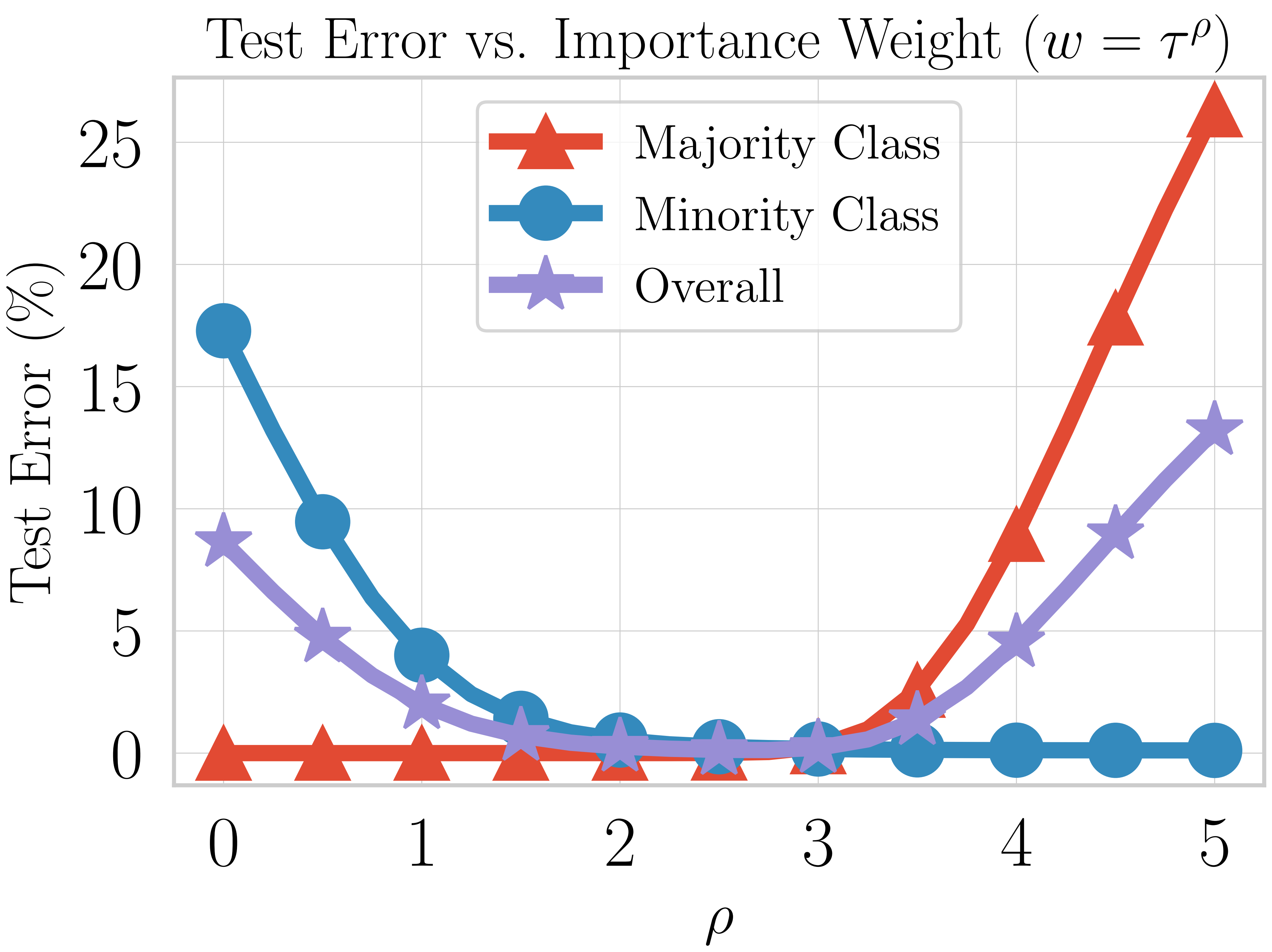
low error when $w = \tau^3$



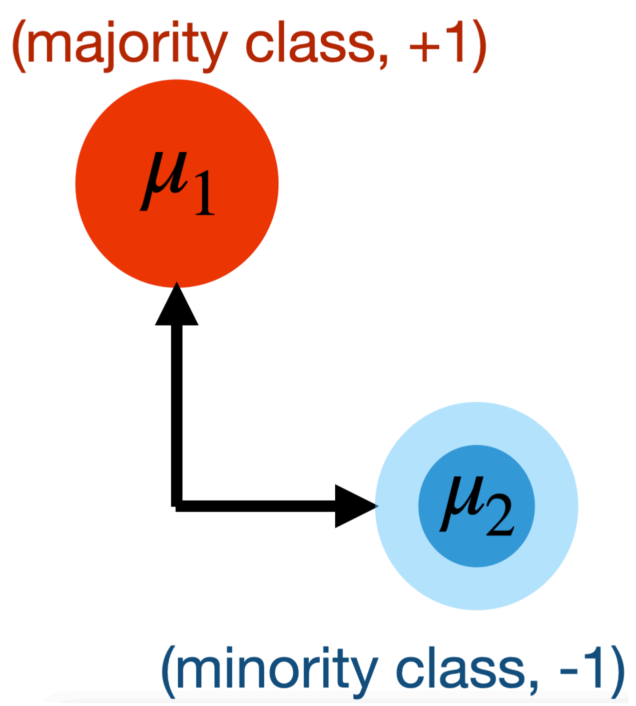
Exponentiate the weights and train on biased loss



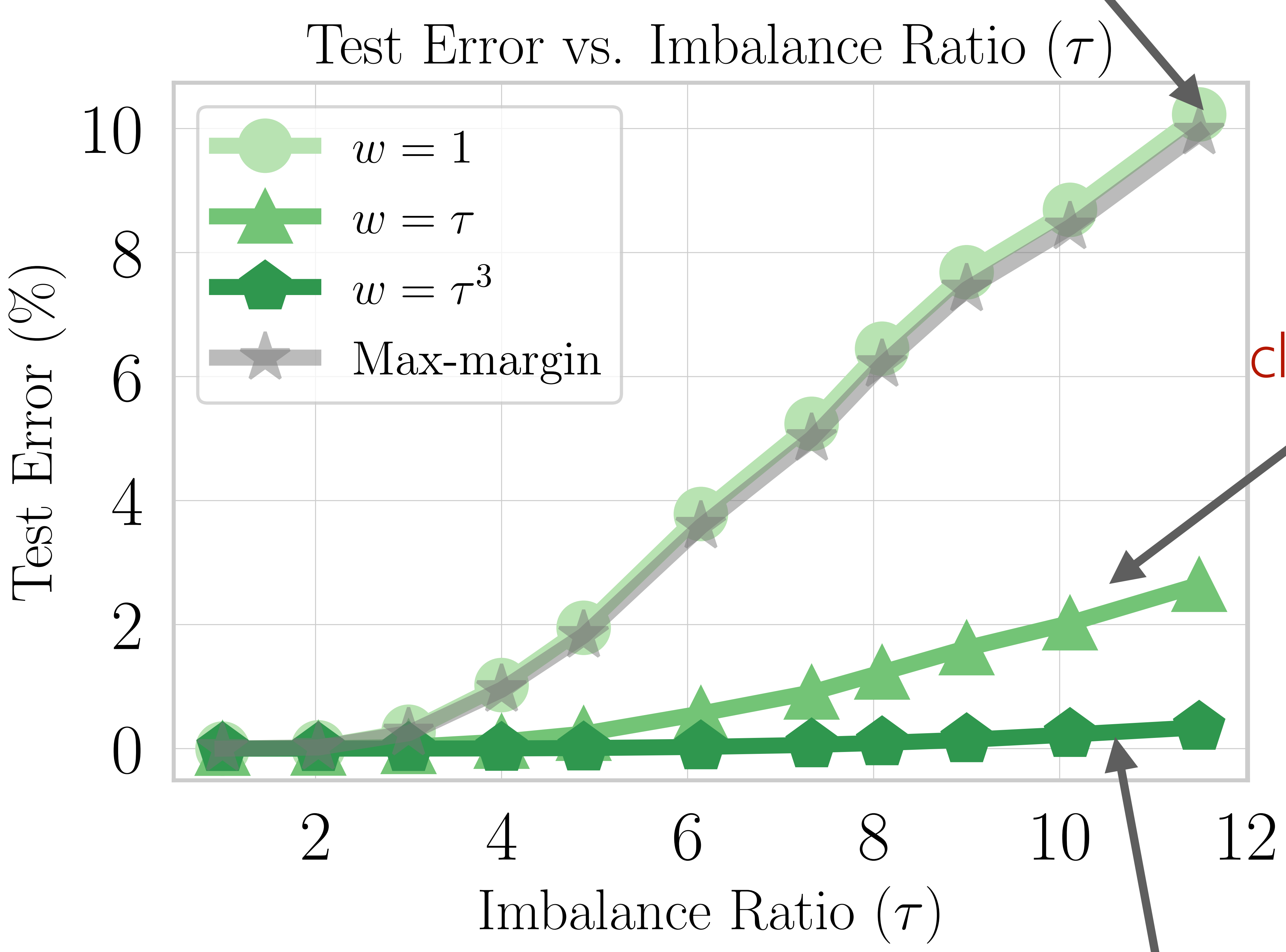




Exponentiate the weights and train on biased loss

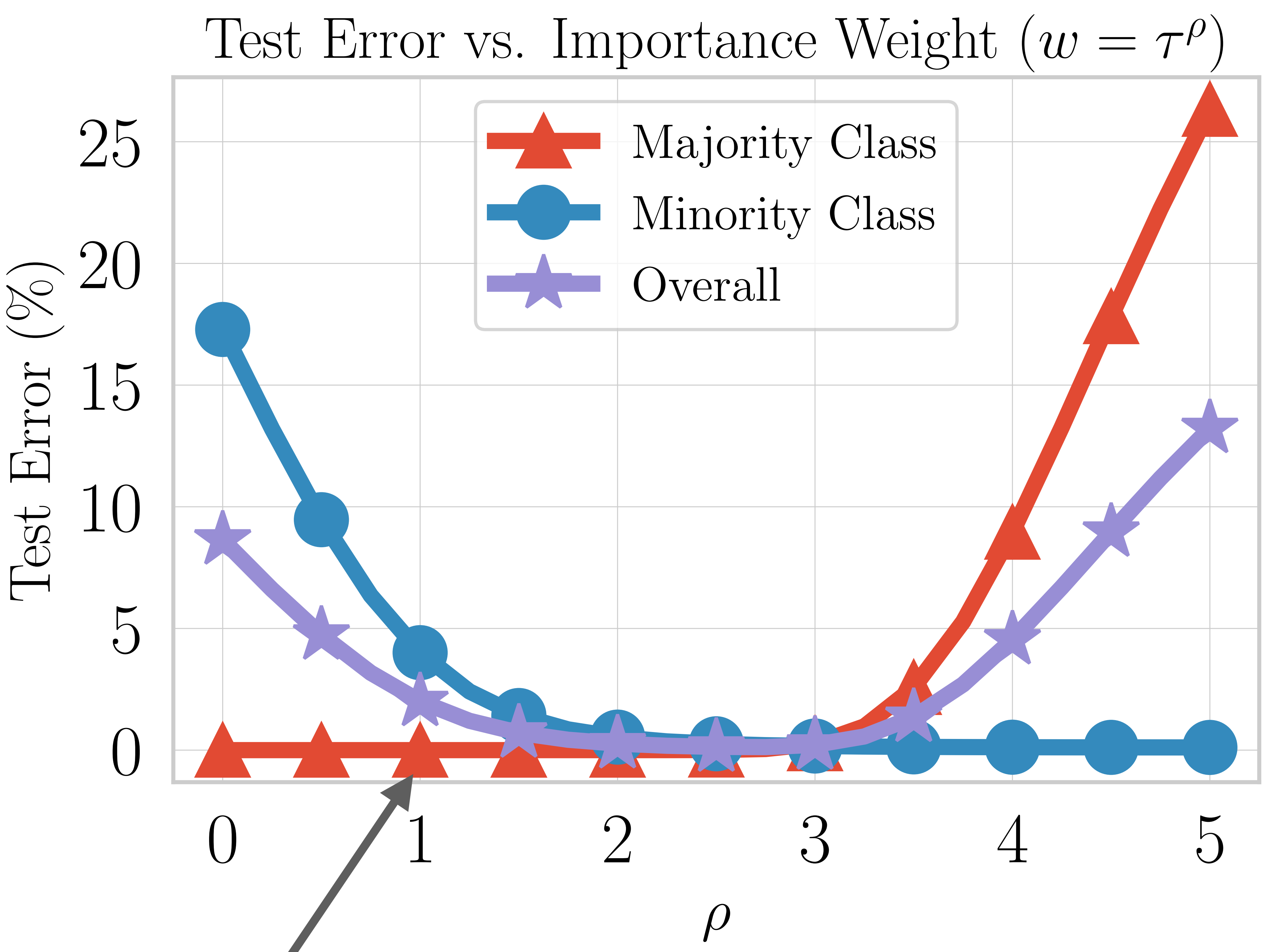


max. margin & no IW



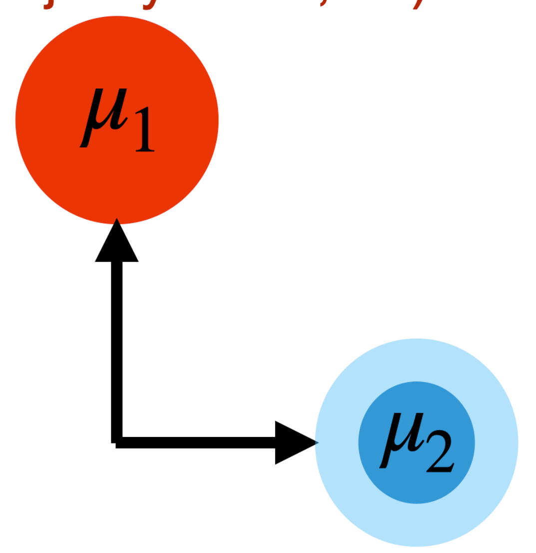
classic IW $w = \tau$

low error when $w = \tau^3$

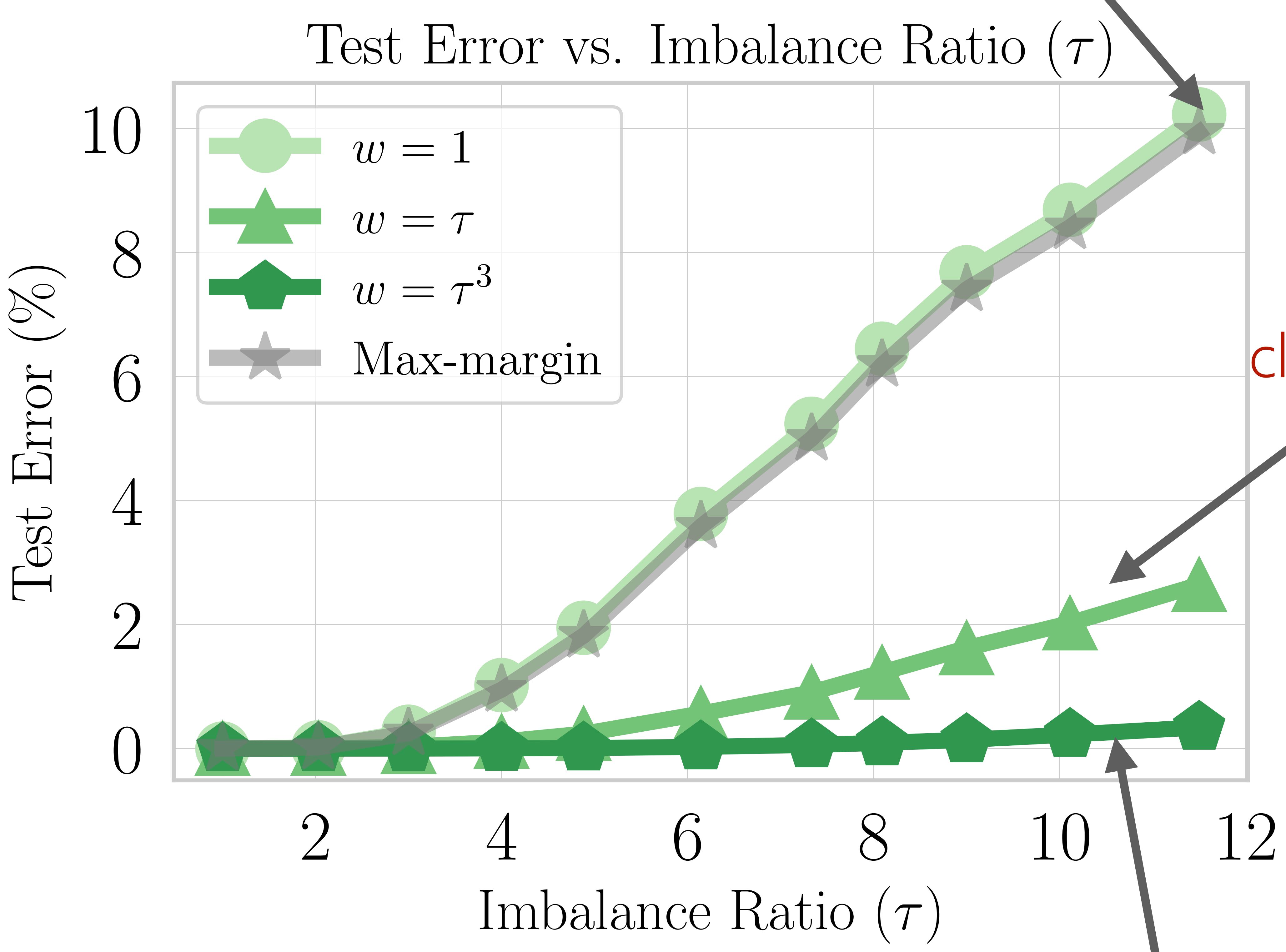


classic IW $w = \tau$

Exponentiate the weights and train on biased loss

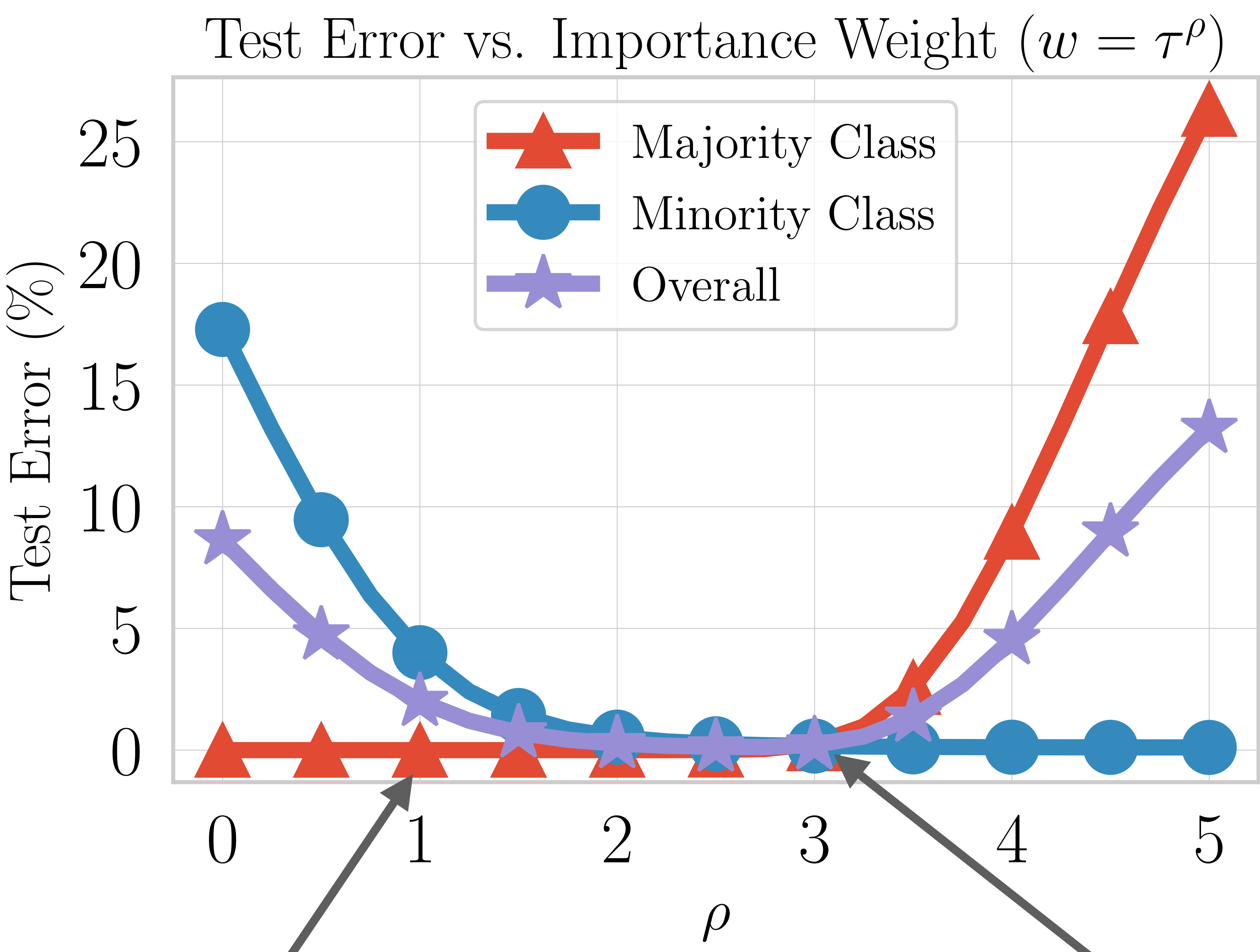


max. margin & no IW



classic IW $w = \tau$

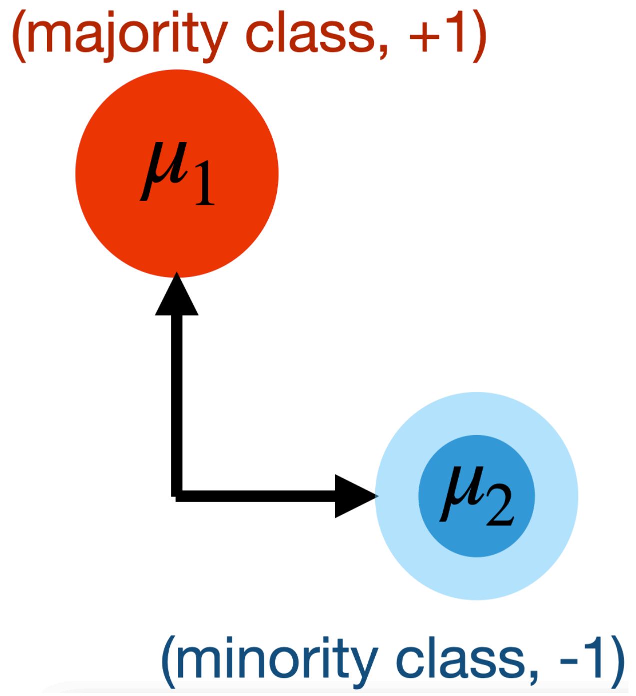
low error when $w = \tau^3$



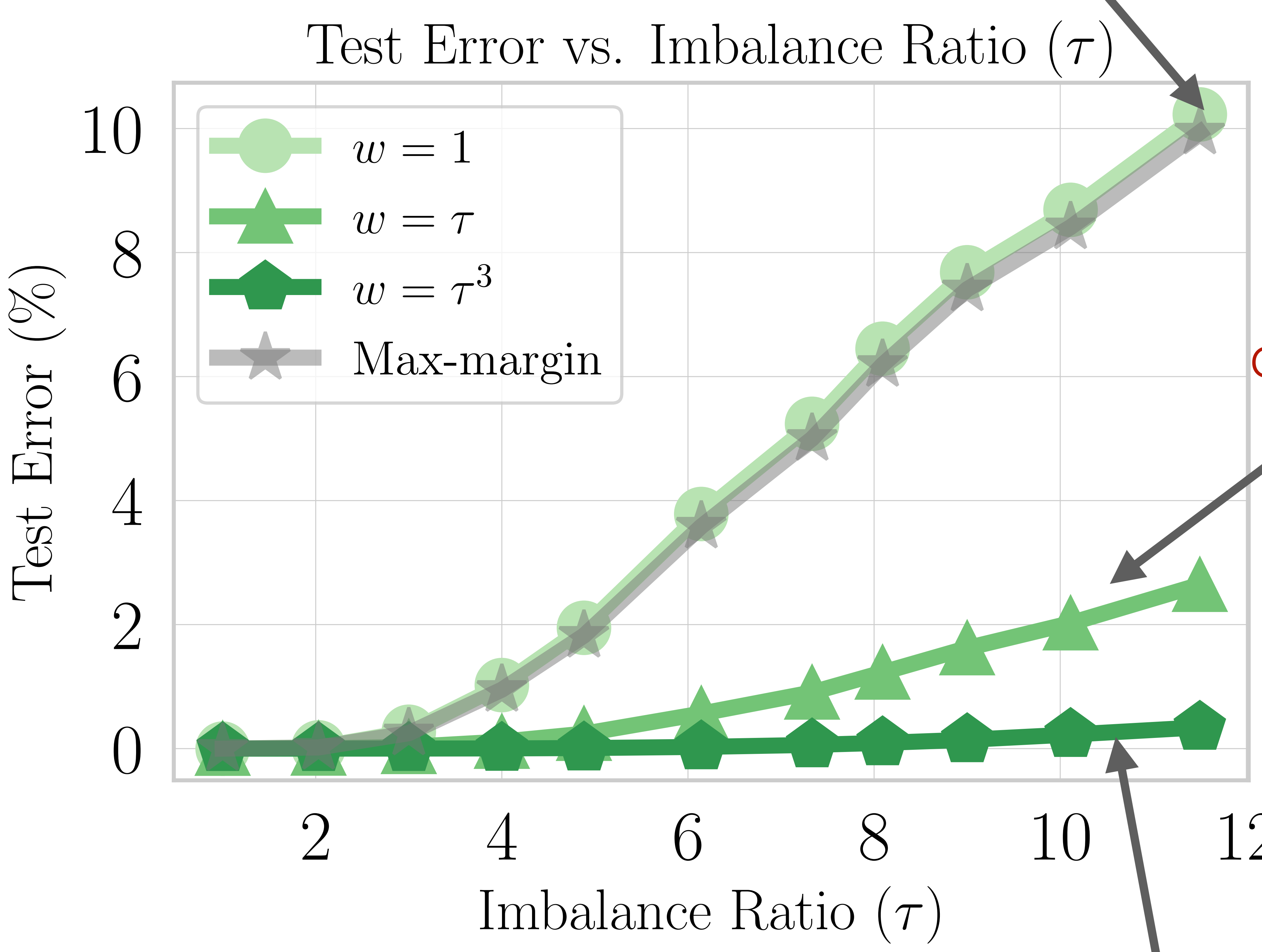
classic IW $w = \tau$

low test error at $w = \tau^3$

Exponentiate the weights and train on biased loss

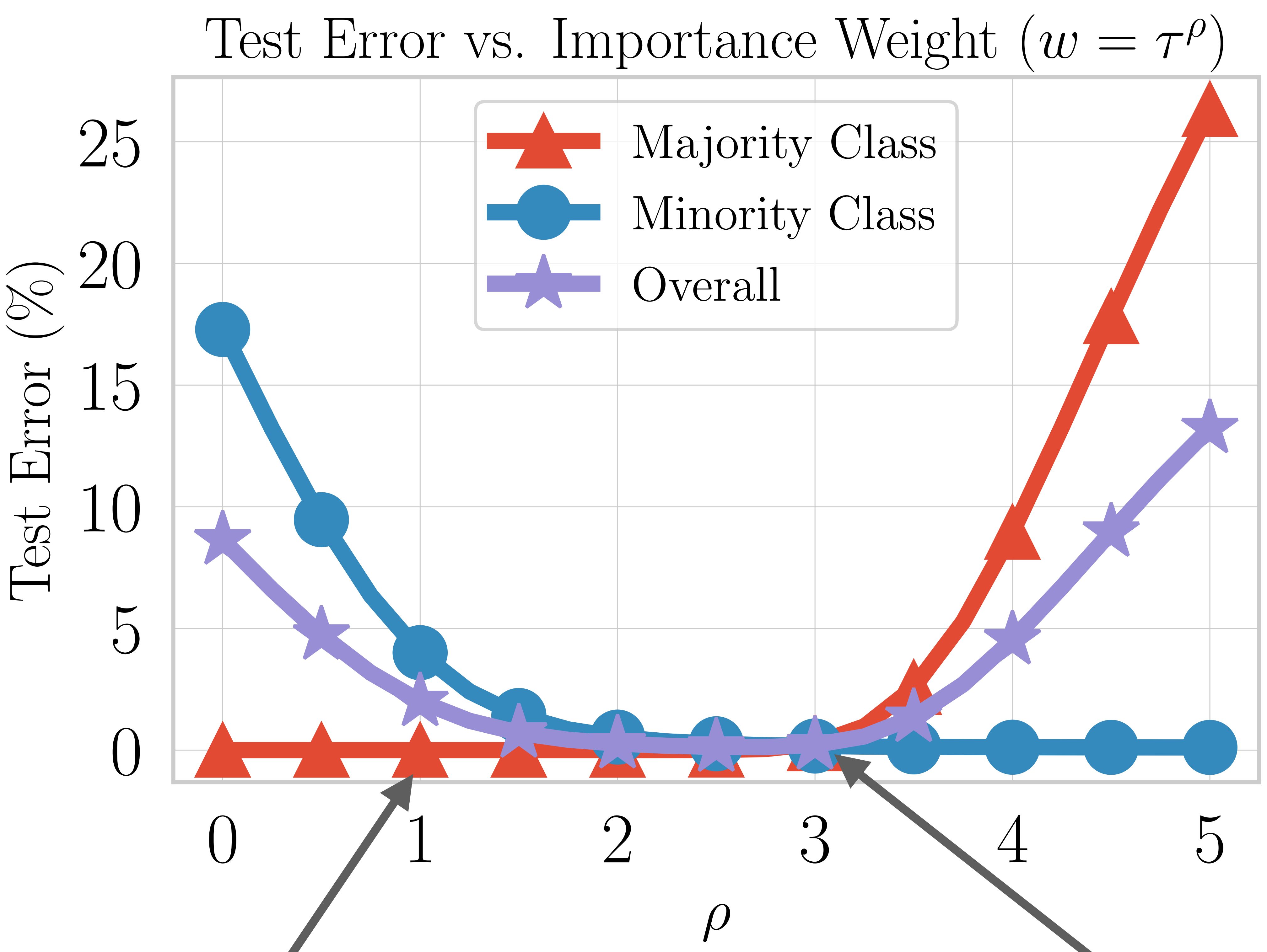


max. margin & no IW



classic IW $w = \tau$

low error when $w = \tau^3$

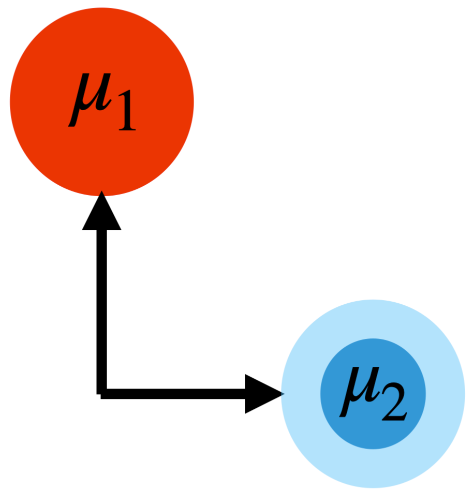


classic IW $w = \tau$

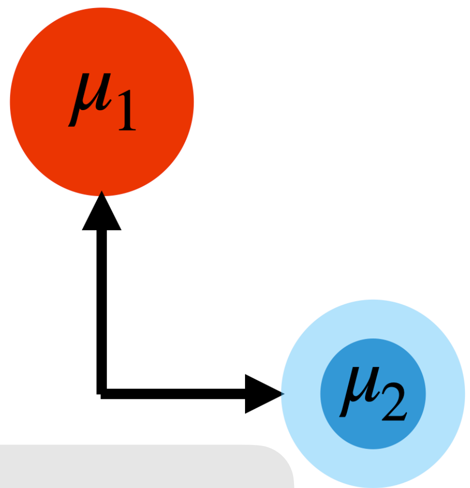
low test error at $w = \tau^3$

In the overparameterized regime, exponentiating weights help!

Proof Idea: Lower Bound the Normalized Margin

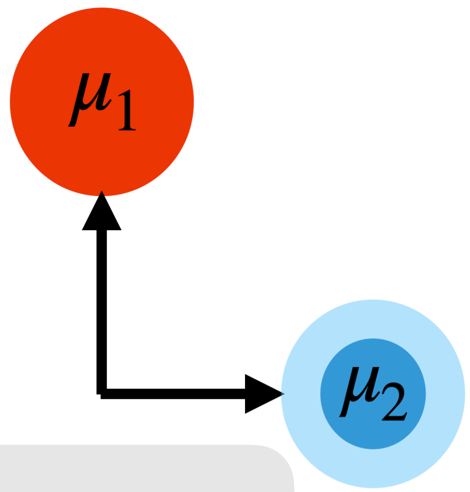


Proof Idea: Lower Bound the Normalized Margin



Step 1: By Hoeffding's inequality

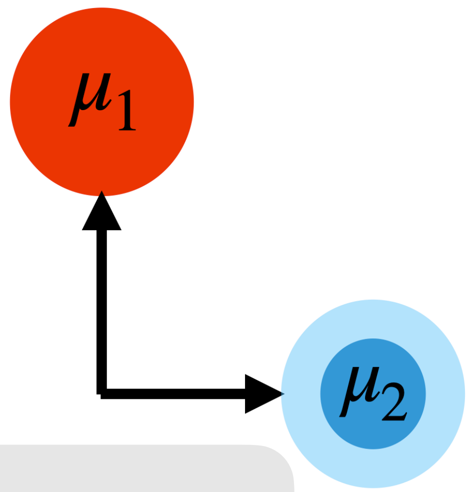
Proof Idea: Lower Bound the Normalized Margin



Step 1: By Hoeffding's inequality

$$\mathrm{TestError}(\theta^{(\infty)}) \leq \frac{1}{2}\left[\exp\left(-\frac{\langle \theta^{(\infty)}, \mu_1 \rangle^2}{\|\theta^{(\infty)}\|^2}\right) + \exp\left(-\frac{\langle \theta^{(\infty)}, \mu_2 \rangle^2}{\|\theta^{(\infty)}\|^2}\right)\right]$$

Proof Idea: Lower Bound the Normalized Margin

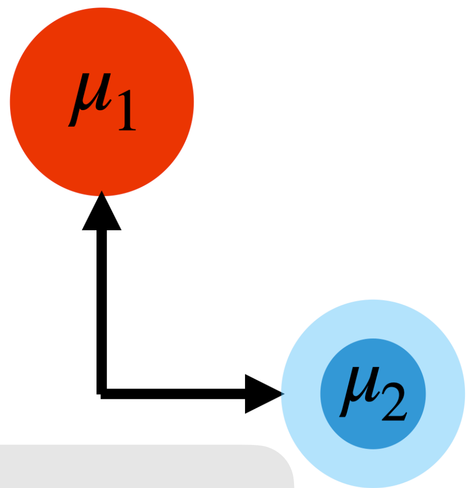


Step 1: By Hoeffding's inequality

$$\mathrm{TestError}(\theta^{(\infty)}) \leq \frac{1}{2}\left[\exp\left(-\frac{\langle \theta^{(\infty)}, \mu_1 \rangle^2}{\|\theta^{(\infty)}\|^2}\right) + \exp\left(-\frac{\langle \theta^{(\infty)}, \mu_2 \rangle^2}{\|\theta^{(\infty)}\|^2}\right)\right]$$

Step 2: Bound on the normalized margins by tracking iterates of GD

Proof Idea: Lower Bound the Normalized Margin



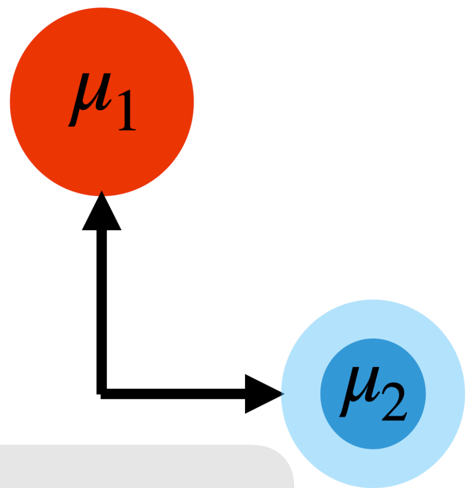
Step 1: By Hoeffding's inequality

$$\mathrm{TestError}(\theta^{(\infty)}) \leq \frac{1}{2}\left[\exp\left(-\frac{\langle \theta^{(\infty)}, \mu_1 \rangle^2}{\|\theta^{(\infty)}\|^2}\right) + \exp\left(-\frac{\langle \theta^{(\infty)}, \mu_2 \rangle^2}{\|\theta^{(\infty)}\|^2}\right)\right]$$

Step 2: Bound on the normalized margins by tracking iterates of GD

$$\frac{\langle \mu_2, \theta^{(t+1)} \rangle}{\|\theta^{(t+1)}\|} \gtrsim \frac{|\mathcal{N}|\|\mu\|^2}{\sqrt{d}}\left[\sum_{s=0}^{t} \frac{\sum_{i \in \mathcal{N}} w(\ell_i^{(s)})^2 - \frac{1}{\|\mu\|}\sum_{j \in \mathcal{P}} (\ell_j^{(s)})^2}{\sum_{k=1}^{n} w_i(\ell_k^{(s)})^2}\right]$$

Proof Idea: Lower Bound the Normalized Margin



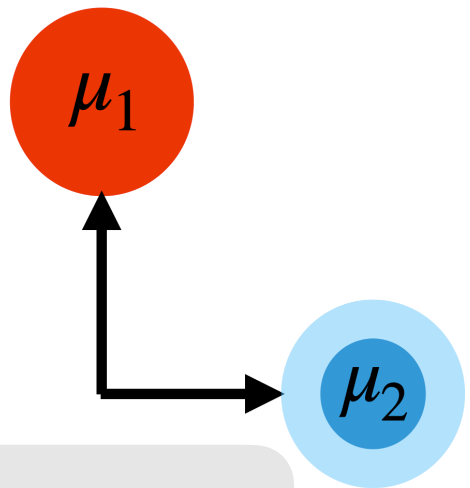
Step 1: By Hoeffding's inequality

$$\mathrm{TestError}(\theta^{(\infty)}) \leq \frac{1}{2}\left[\exp\left(-\frac{\langle \theta^{(\infty)}, \mu_1\rangle^2}{\|\theta^{(\infty)}\|^2}\right) + \exp\left(-\frac{\langle \theta^{(\infty)}, \mu_2\rangle^2}{\|\theta^{(\infty)}\|^2}\right)\right]$$

Step 2: Bound on the normalized margins by tracking iterates of GD

$$\frac{\langle \mu_2, \theta^{(t+1)}\rangle}{\|\theta^{(t+1)}\|} \gtrsim \frac{|\mathcal{N}|\|\mu\|^2}{\sqrt{d}}\left[\sum_{s=0}^{t} \frac{\sum_{i\in\mathcal{N}} w(\ell_i^{(s)})^2 - \frac{1}{\|\mu\|}\sum_{j\in\mathcal{P}}(\ell_j^{(s)})^2}{\sum_{k=1}^{n} w_i(\ell_k^{(s)})^2}\right]$$

Proof Idea: Lower Bound the Normalized Margin



Step 1: By Hoeffding's inequality

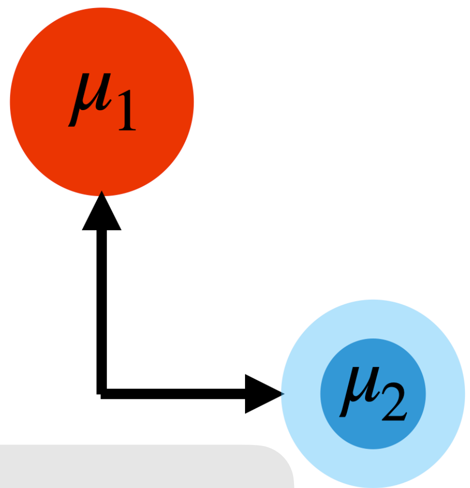
$$\text{TestError}(\theta^{(\infty)}) \le \frac{1}{2}\left[\exp\left(-\frac{\langle \theta^{(\infty)}, \mu_1\rangle^2}{\|\theta^{(\infty)}\|^2}\right) + \exp\left(-\frac{\langle \theta^{(\infty)}, \mu_2\rangle^2}{\|\theta^{(\infty)}\|^2}\right)\right]$$

Step 2: Bound on the normalized margins by tracking iterates of GD

$$\frac{\langle \mu_2, \theta^{(t+1)}\rangle}{\|\theta^{(t+1)}\|} \gtrsim \frac{|\mathcal{N}|\|\mu\|^2}{\sqrt{d}}\left[\sum_{s=0}^{t} \frac{\sum_{i\in\mathcal{N}} w(\ell_i^{(s)})^2 - \frac{1}{\|\mu\|}\sum_{j\in\mathcal{P}}(\ell_j^{(s)})^2}{\sum_{k=1}^{n} w_i(\ell_k^{(s)})^2}\right]$$

- We are done if this is a positive constant

Proof Idea: Lower Bound the Normalized Margin



Step 1: By Hoeffding's inequality

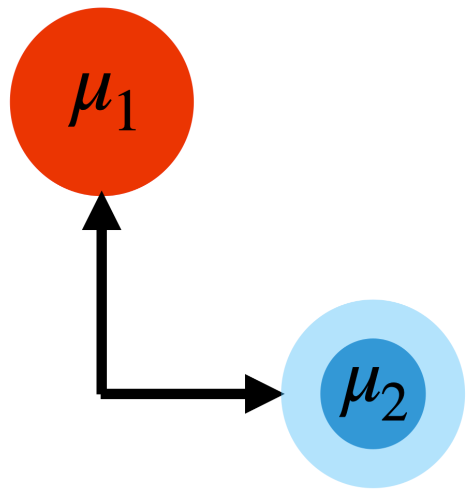
$$\text{TestError}(\theta^{(\infty)}) \leq \frac{1}{2}\left[\exp\left(-\frac{\langle \theta^{(\infty)}, \mu_1 \rangle^2}{\|\theta^{(\infty)}\|^2}\right) + \exp\left(-\frac{\langle \theta^{(\infty)}, \mu_2 \rangle^2}{\|\theta^{(\infty)}\|^2}\right)\right]$$

Step 2: Bound on the normalized margins by tracking iterates of GD

$$\frac{\langle \mu_2, \theta^{(t+1)} \rangle}{\|\theta^{(t+1)}\|} \gtrsim \frac{|\mathcal{N}|\|\mu\|^2}{\sqrt{d}}\left[\sum_{s=0}^{t} \frac{\sum_{i\in\mathcal{N}} w(\ell_i^{(s)})^2 - \frac{1}{\|\mu\|}\sum_{j\in\mathcal{P}} (\ell_j^{(s)})^2}{\sum_{k=1}^{n} w_i(\ell_k^{(s)})^2}\right]$$

- We are done if this is a positive constant
- Need to show numerator is lower bounded

Loss Ratio Bound

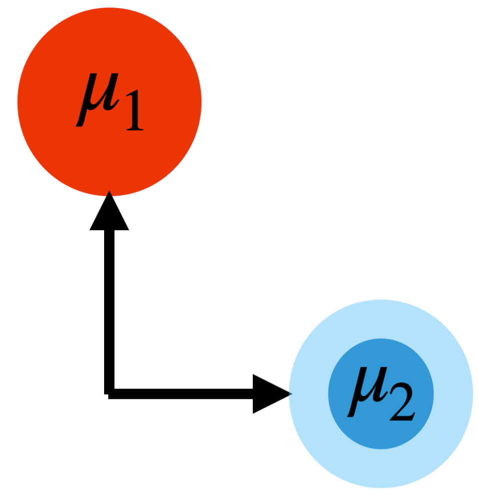


Step 2: Bound on the normalized margins by tracking iterates of GD

$$\frac{\langle \mu_2, \theta^{(t+1)}\rangle}{\|\theta^{(t+1)}\|} \gtrsim \frac{\|\mu\|^2|\mathcal{N}|}{\sqrt{d}}\left[\sum_{s=0}^{t} \frac{\sum_{i\in\mathcal{N}} w(\ell_i^{(s)})^2 - \frac{1}{\|\mu\|}\sum_{j\in\mathcal{P}}(\ell_j^{(s)})^2}{\sum_{k=1}^{n} w_i(\ell_k^{(s)})^2}\right]$$

- Need to show numerator is lower bounded

Loss Ratio Bound



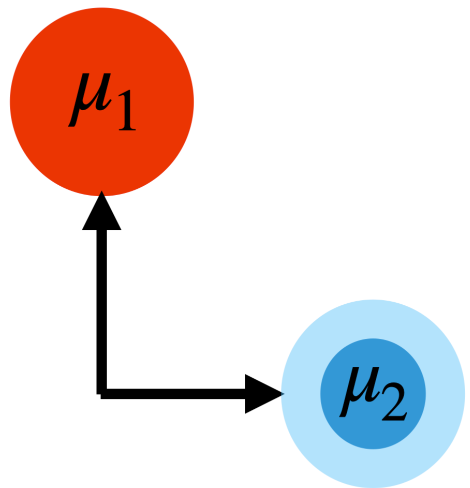
Step 2: Bound on the normalized margins by tracking iterates of GD

$$\frac{\langle \mu_2, \theta^{(t+1)} \rangle}{\|\theta^{(t+1)}\|} \gtrsim \frac{\|\mu\|^2 |\mathcal{N}|}{\sqrt{d}} \left[\sum_{s=0}^{t} \frac{\sum_{i \in \mathcal{N}} w(\ell_i^{(s)})^2 - \frac{1}{\|\mu\|} \sum_{j \in \mathcal{P}} (\ell_j^{(s)})^2}{\sum_{k=1}^{n} w_i (\ell_k^{(s)})^2} \right]$$

- Need to show numerator is lower bounded

Loss Ratio Bound: If the step-size is small enough, for all $s \in \{1, \ldots\}$

Loss Ratio Bound



Step 2: Bound on the normalized margins by tracking iterates of GD

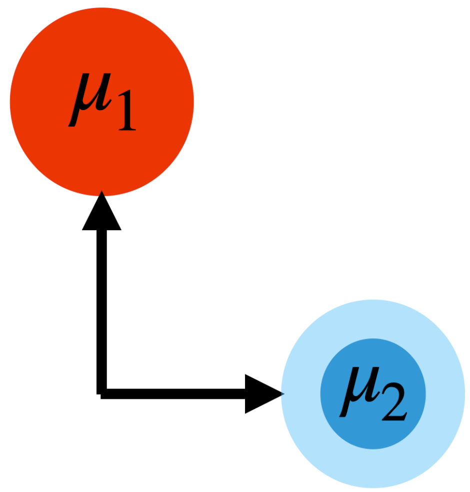
$$\frac{\langle \mu_2, \theta^{(t+1)} \rangle}{\|\theta^{(t+1)}\|} \gtrsim \frac{\|\mu\|^2 |\mathcal{N}|}{\sqrt{d}} \left[\sum_{s=0}^{t} \frac{\sum_{i \in \mathcal{N}} w(\ell_i^{(s)})^2 - \frac{1}{\|\mu\|} \sum_{j \in \mathcal{P}} (\ell_j^{(s)})^2}{\sum_{k=1}^{n} w_i (\ell_k^{(s)})^2} \right]$$

- Need to show numerator is lower bounded

Loss Ratio Bound: If the step-size is small enough, for all $s \in \{1, \ldots\}$

For all $i \neq j$: $$\frac{\ell_i^{(s)}}{\ell_j^{(s)}} \lesssim \left(\frac{w_j}{w_i} \right)^{1/3}$$

Loss Ratio Bound



Step 2: Bound on the normalized margins by tracking iterates of GD

$$\frac{\langle \mu_2, \theta^{(t+1)} \rangle}{\|\theta^{(t+1)}\|} \gtrsim \frac{\|\mu\|^2 |\mathcal{N}|}{\sqrt{d}} \left[\sum_{s=0}^{t} \frac{\sum_{i \in \mathcal{N}} w(\ell_i^{(s)})^2 - \frac{1}{\|\mu\|} \sum_{j \in \mathcal{P}} (\ell_j^{(s)})^2}{\sum_{k=1}^{n} w_i (\ell_k^{(s)})^2} \right] \geq c \cdot \text{denominator}$$

- Need to show numerator is lower bounded

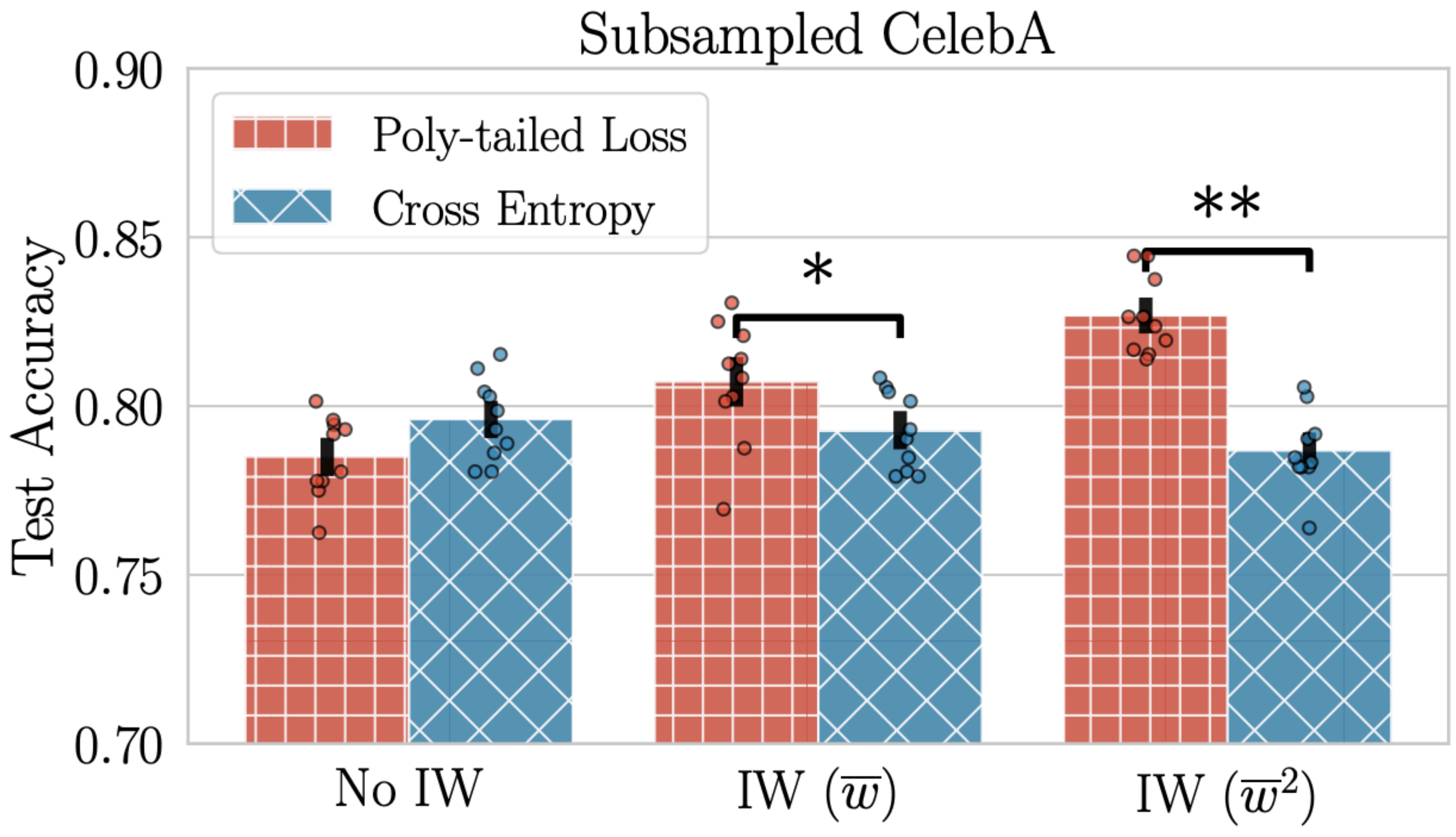
Loss Ratio Bound: If the step-size is small enough, for all $s \in \{1, \ldots\}$

For all $i \neq j$: $$\frac{\ell_i^{(s)}}{\ell_j^{(s)}} \lesssim \left(\frac{w_j}{w_i} \right)^{1/3}$$

Experiments with Neural Network Classifiers

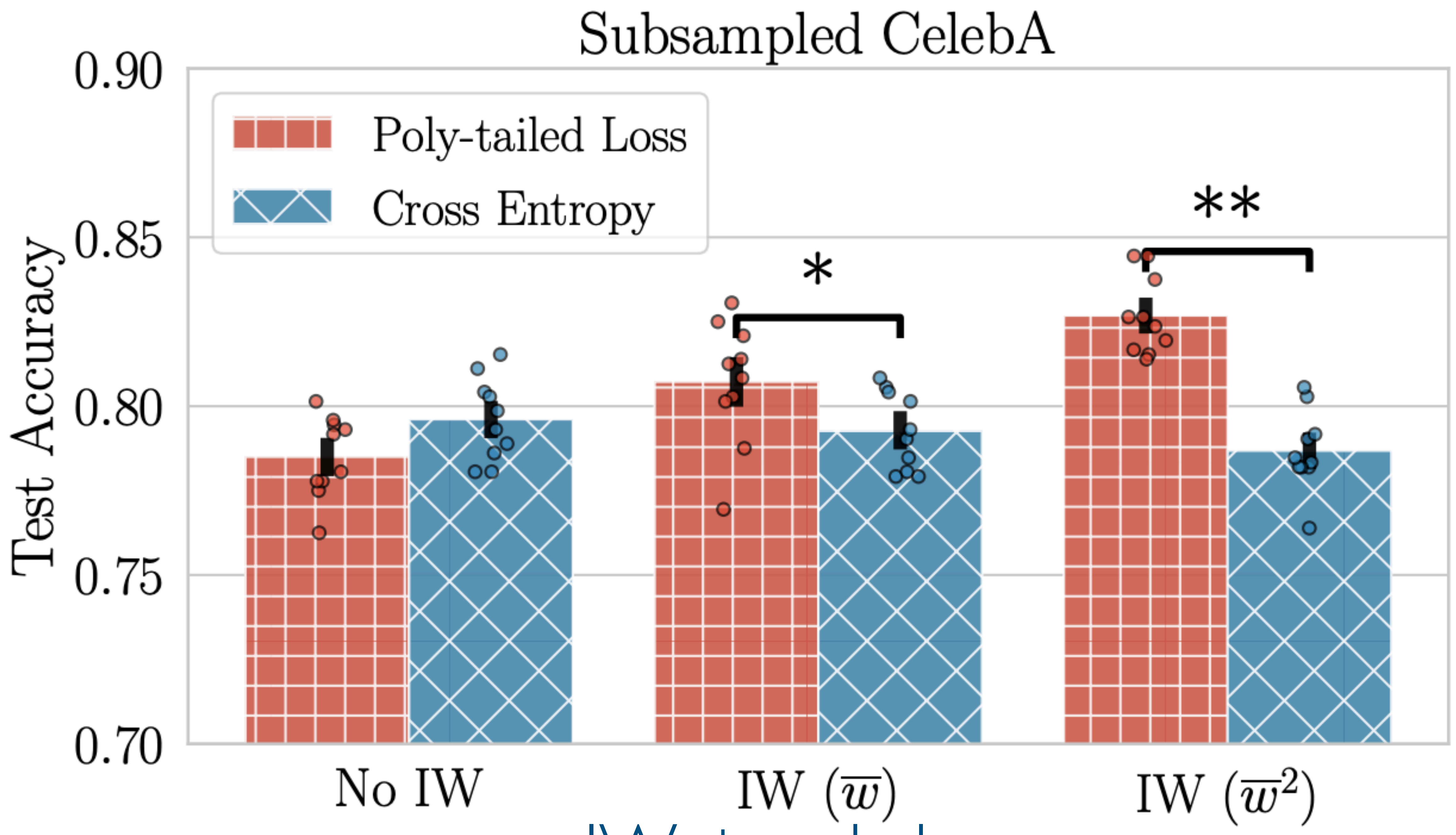
Experiments with Neural Network Classifiers

Interpolating models



Experiments with Neural Network Classifiers

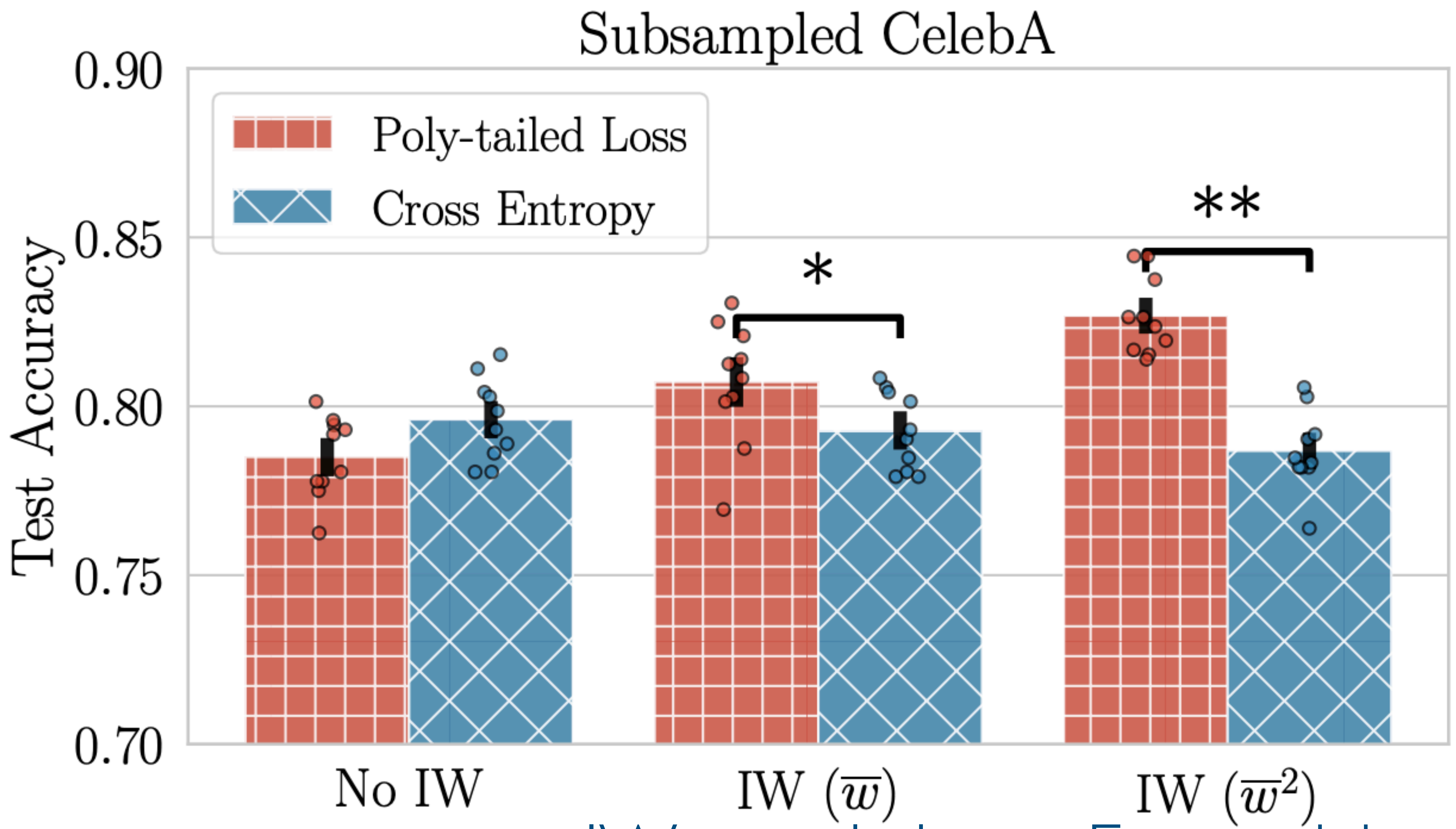
Interpolating models



IW + poly loss improves acc.

Experiments with Neural Network Classifiers

Interpolating models

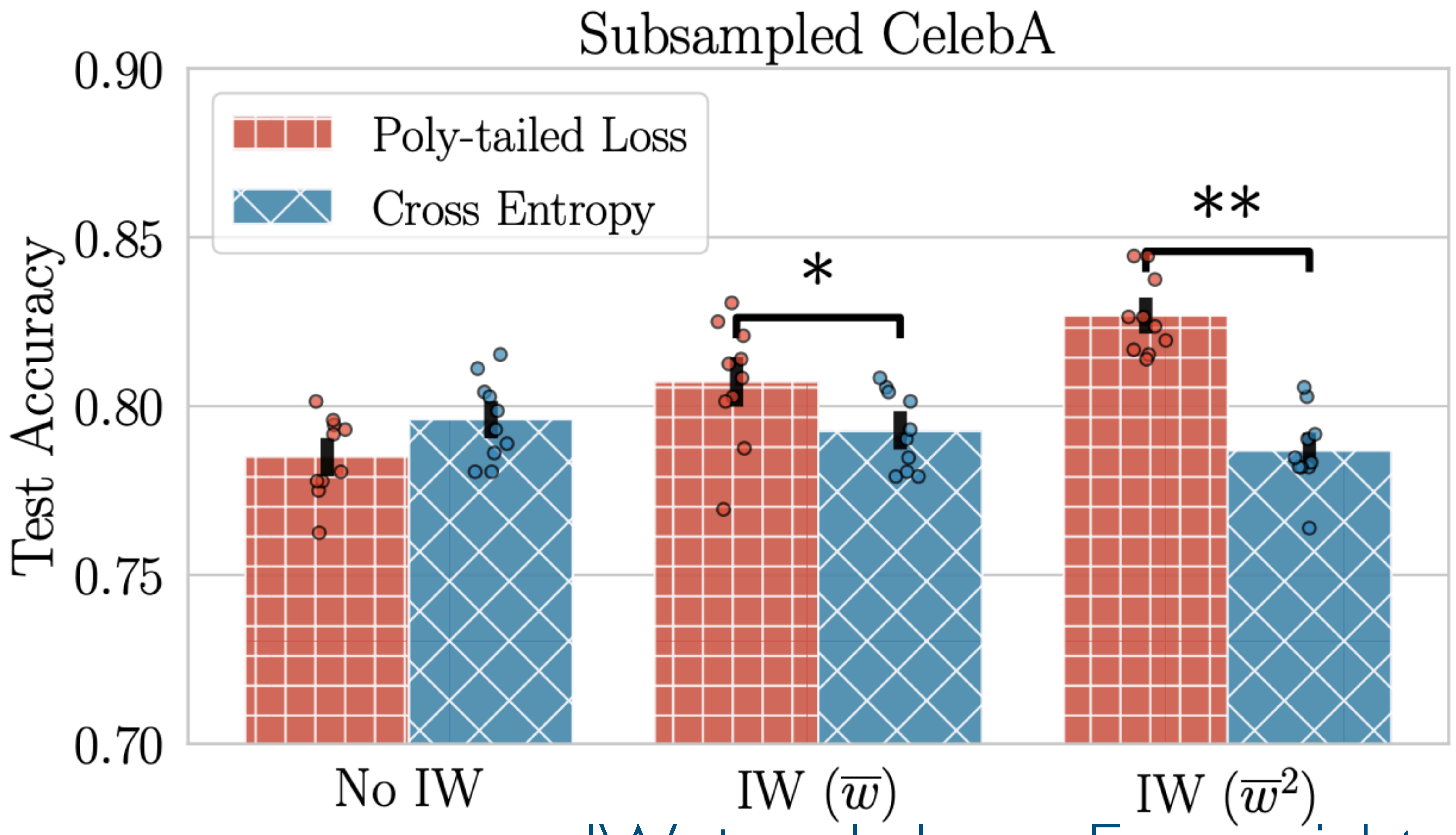


IW + poly loss improves acc.

Exp. weights improves acc.

Experiments with Neural Network Classifiers

Interpolating models



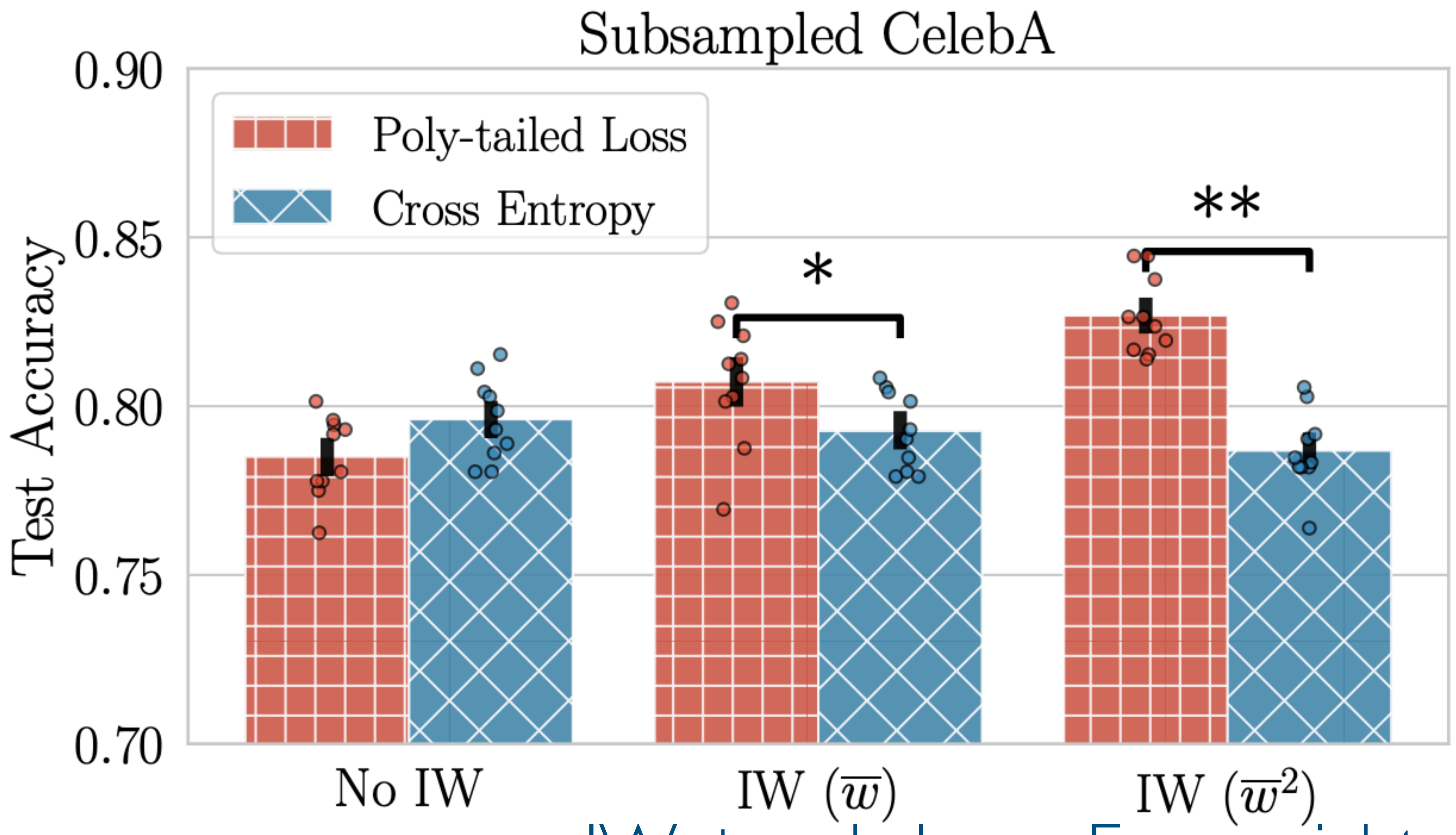
IW + poly loss improves acc.

Exp. weights improves acc.

Polynomial Losses + exponentiated weights improve performance for NNs

Experiments with Neural Network Classifiers

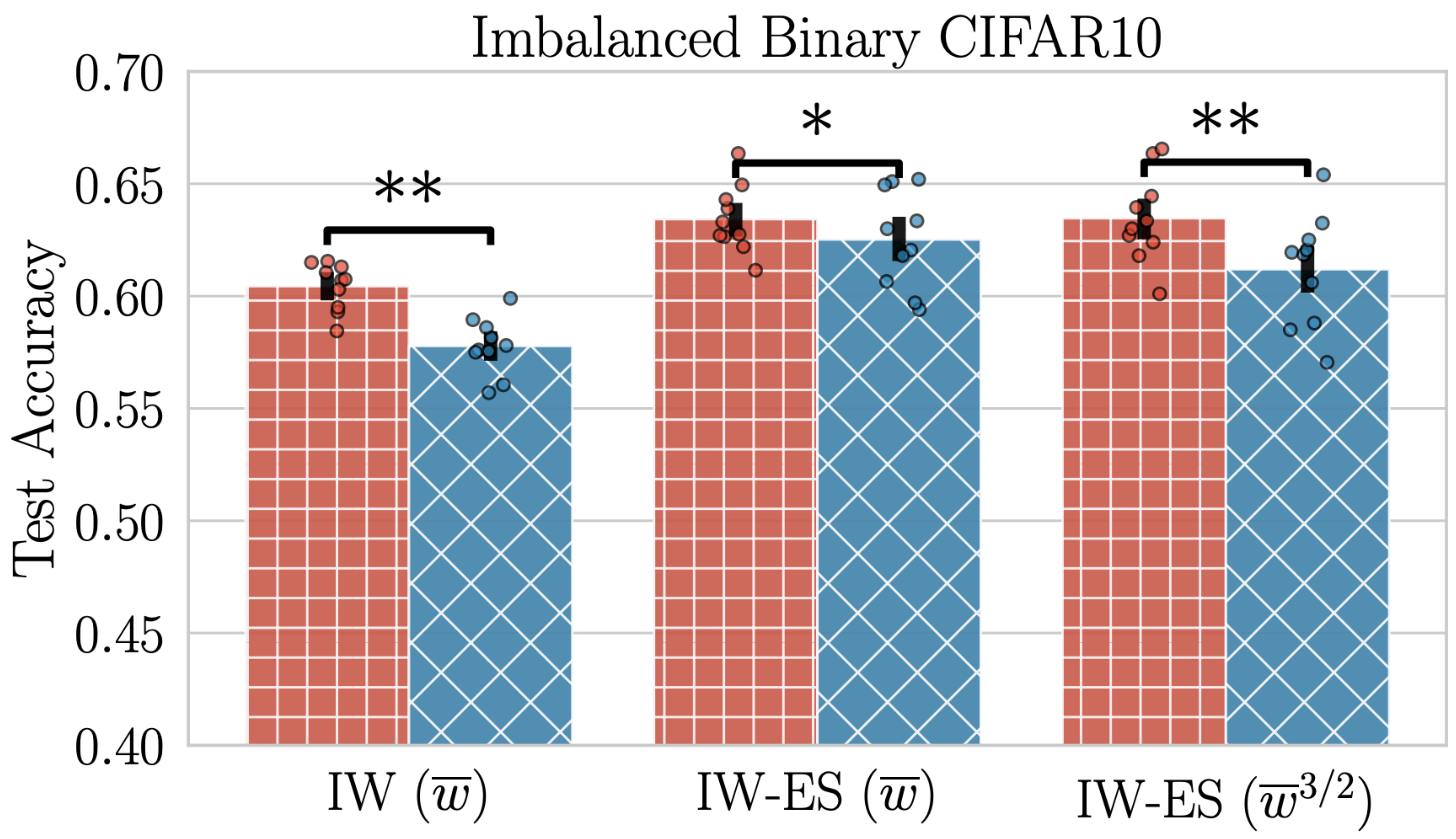
Interpolating models



IW + poly loss improves acc.

Exp. weights improves acc.

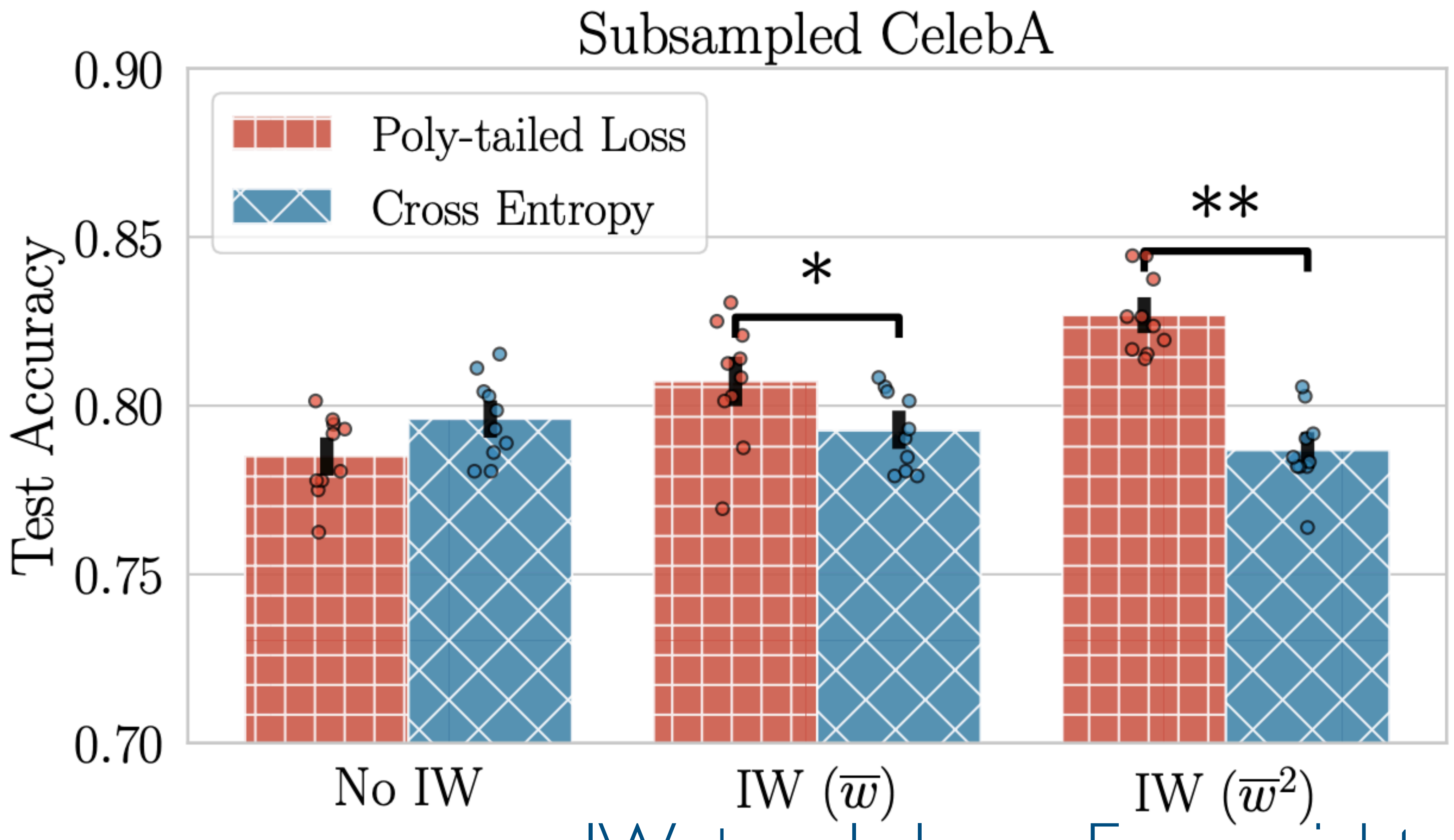
Early Stopping



Polynomial Losses + exponentiated weights improve performance for NNs

Experiments with Neural Network Classifiers

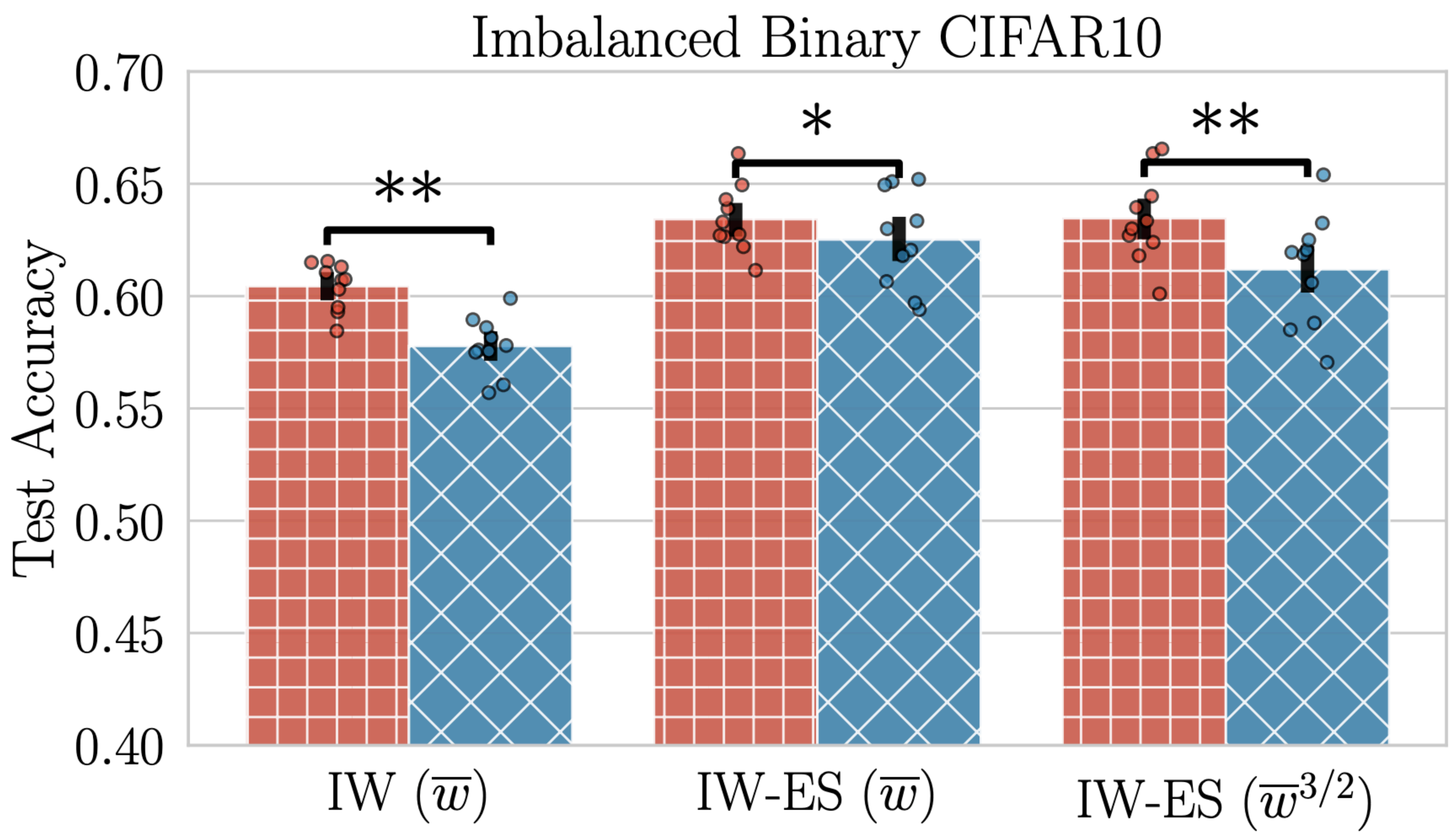
Interpolating models



IW + poly loss improves acc.

Exp. weights improves acc.

Early Stopping

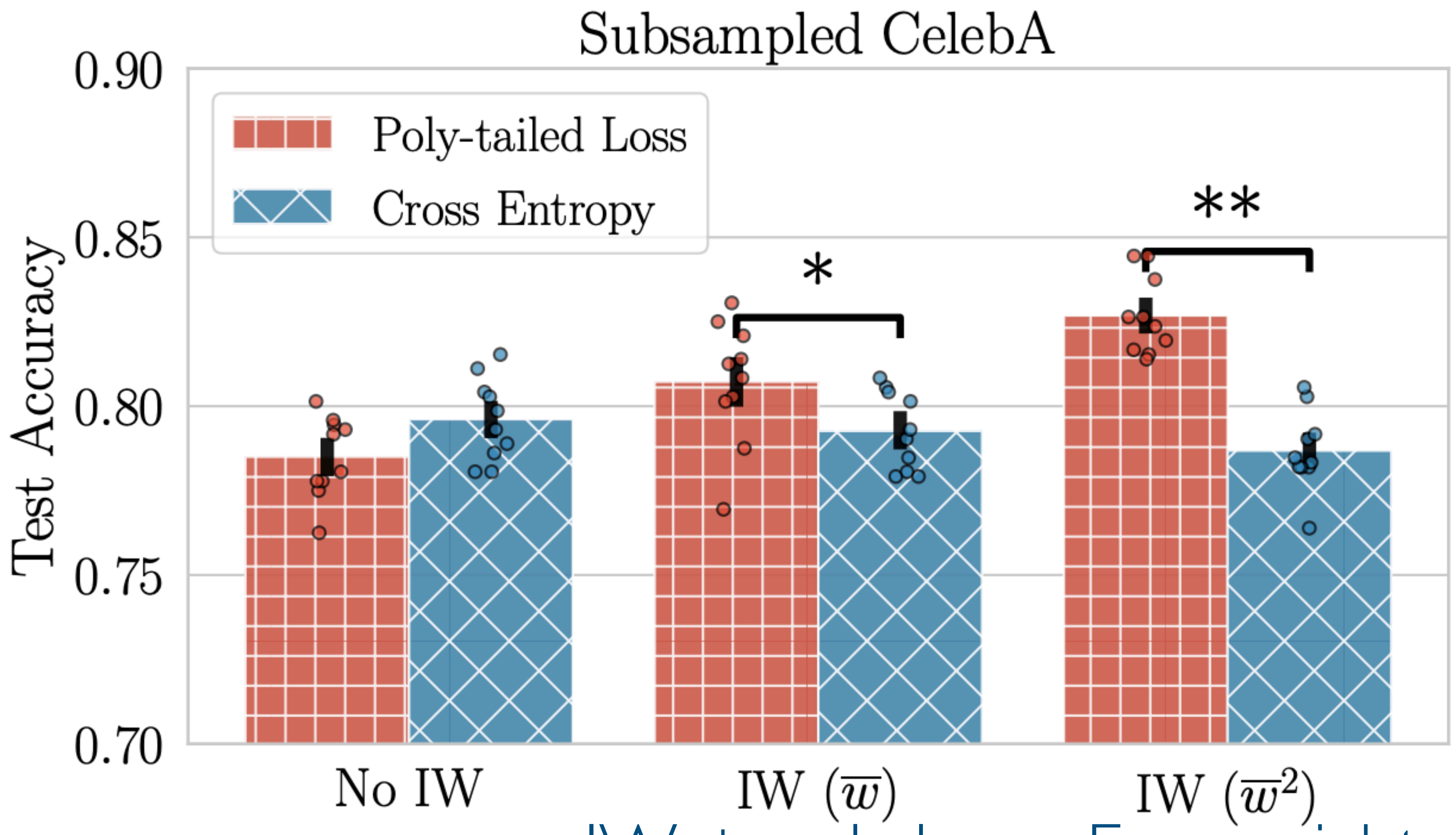


classic IW

Polynomial Losses + exponentiated weights improve performance for NNs

Experiments with Neural Network Classifiers

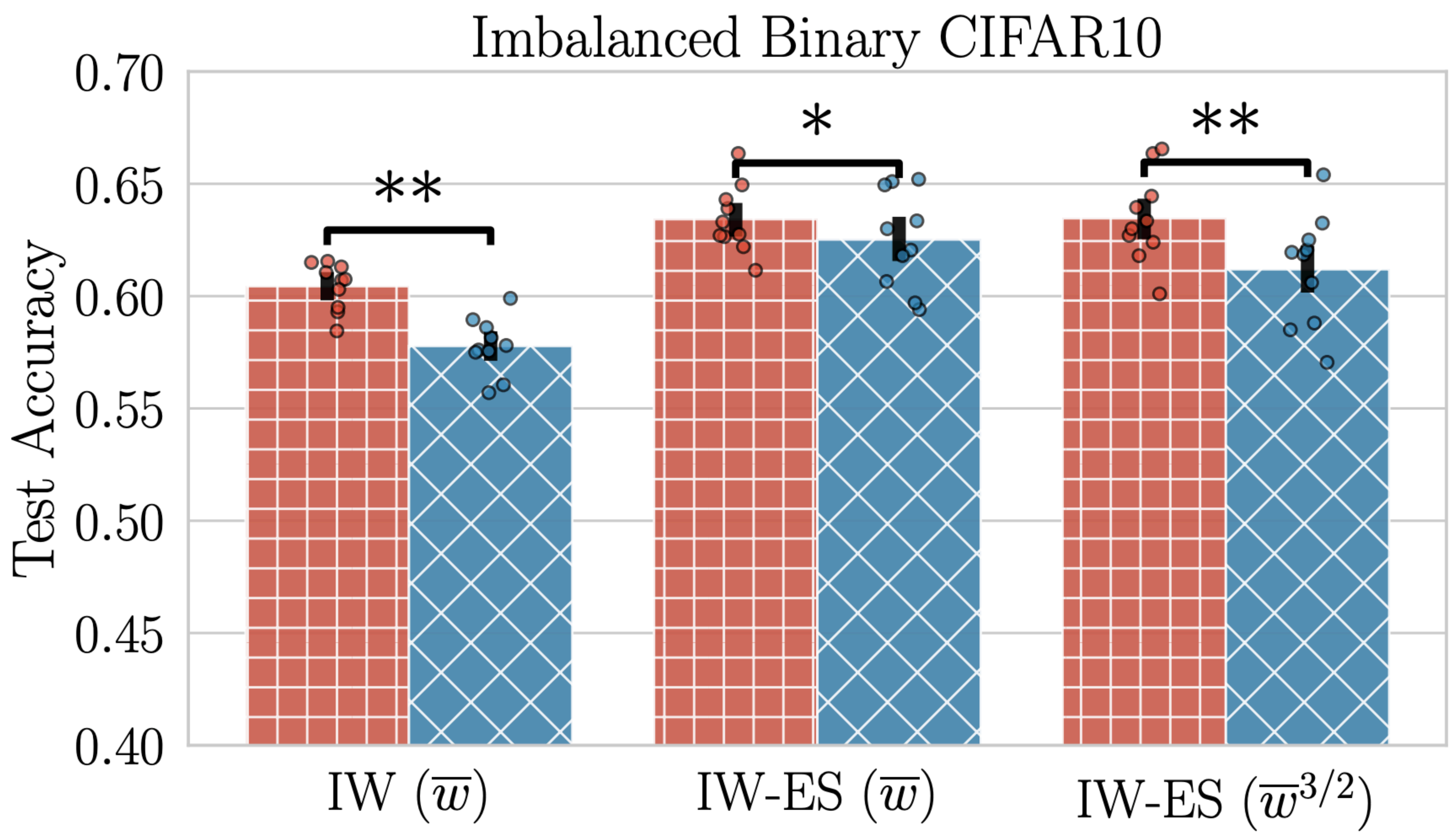
Interpolating models



IW + poly loss improves acc.

Exp. weights improves acc.

Early Stopping



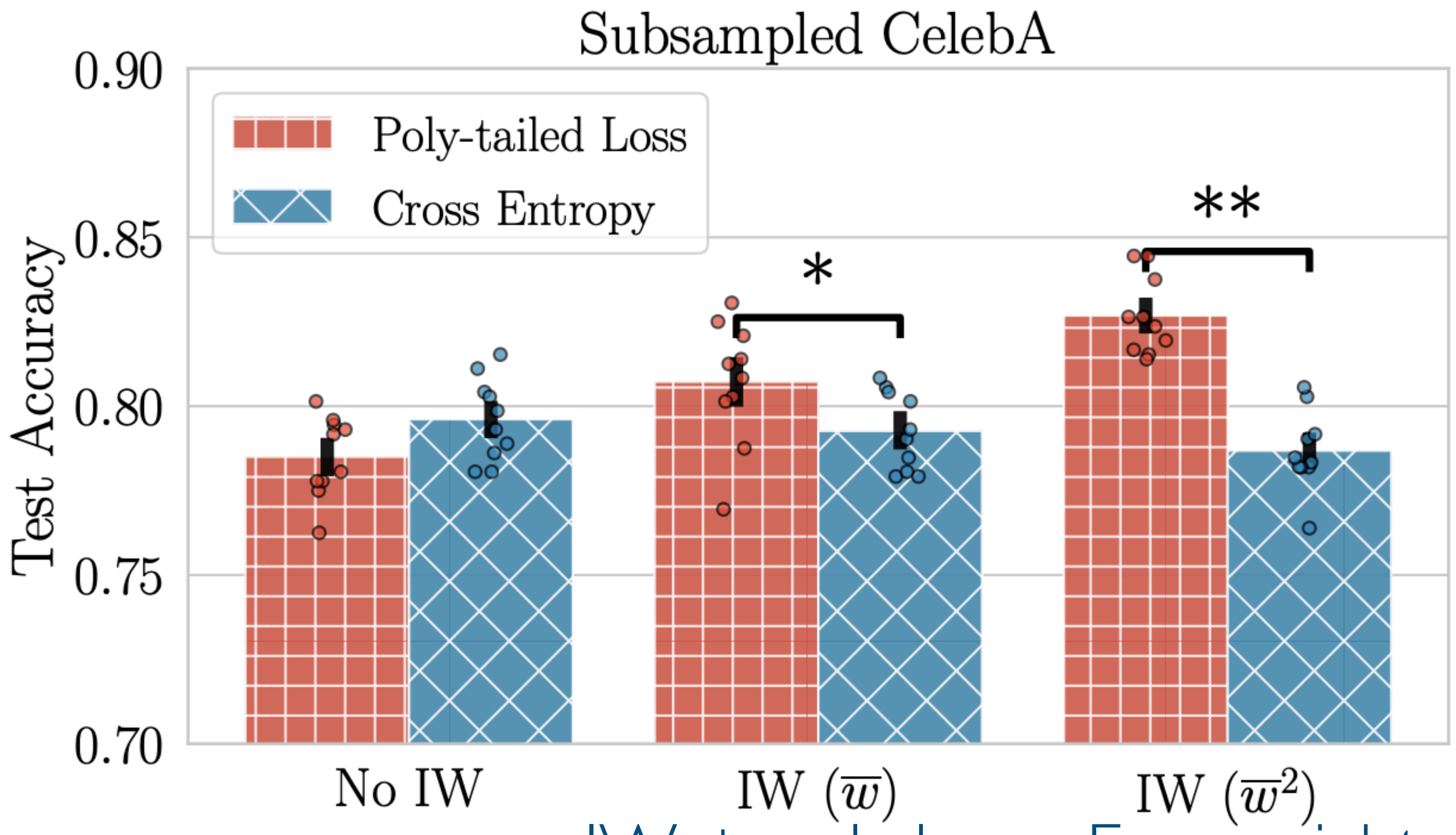
classic IW

exp. weights

Polynomial Losses + exponentiated weights improve performance for NNs

Experiments with Neural Network Classifiers

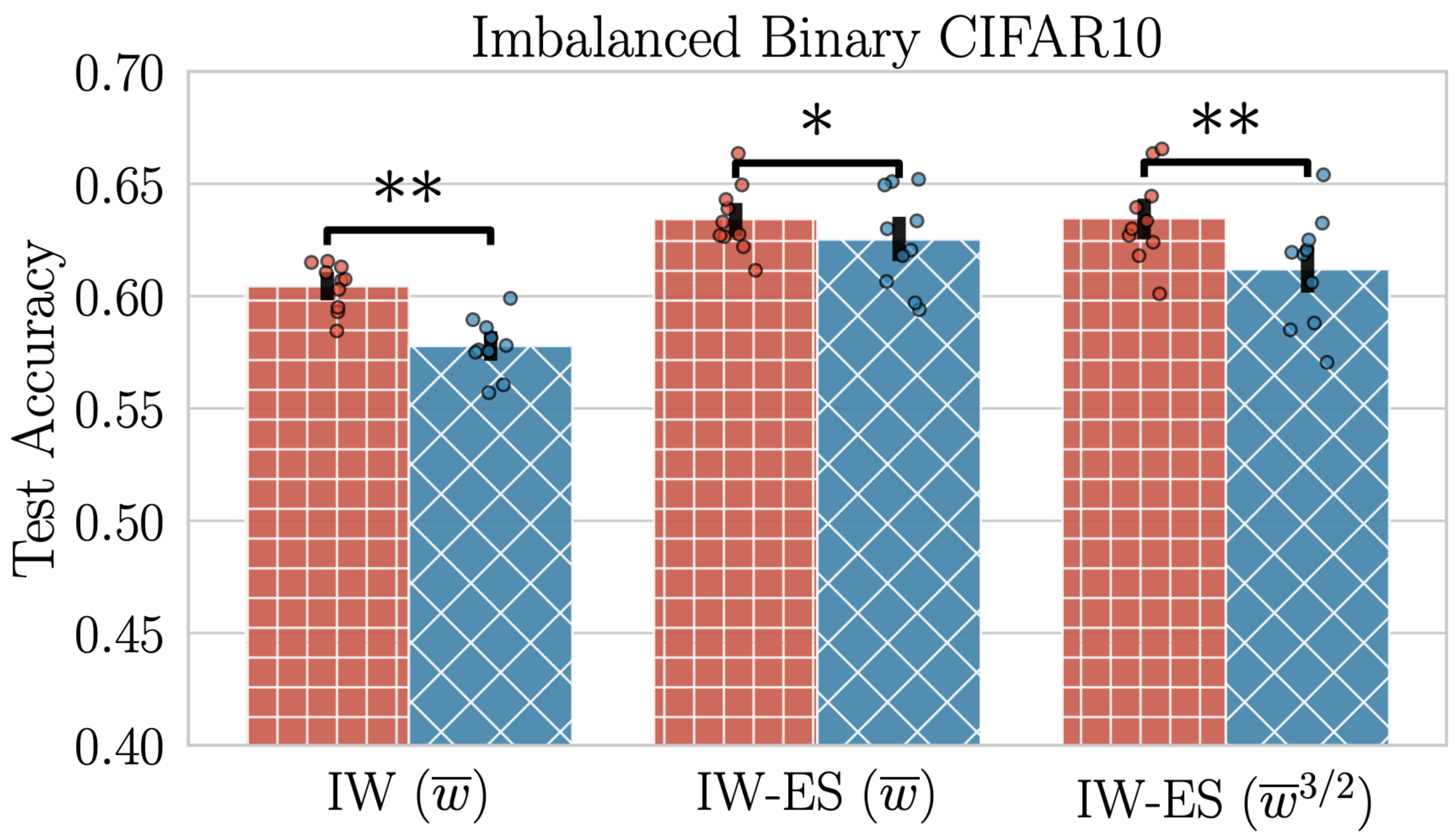
Interpolating models



IW + poly loss improves acc.

Exp. weights improves acc.

Early Stopping



classic IW

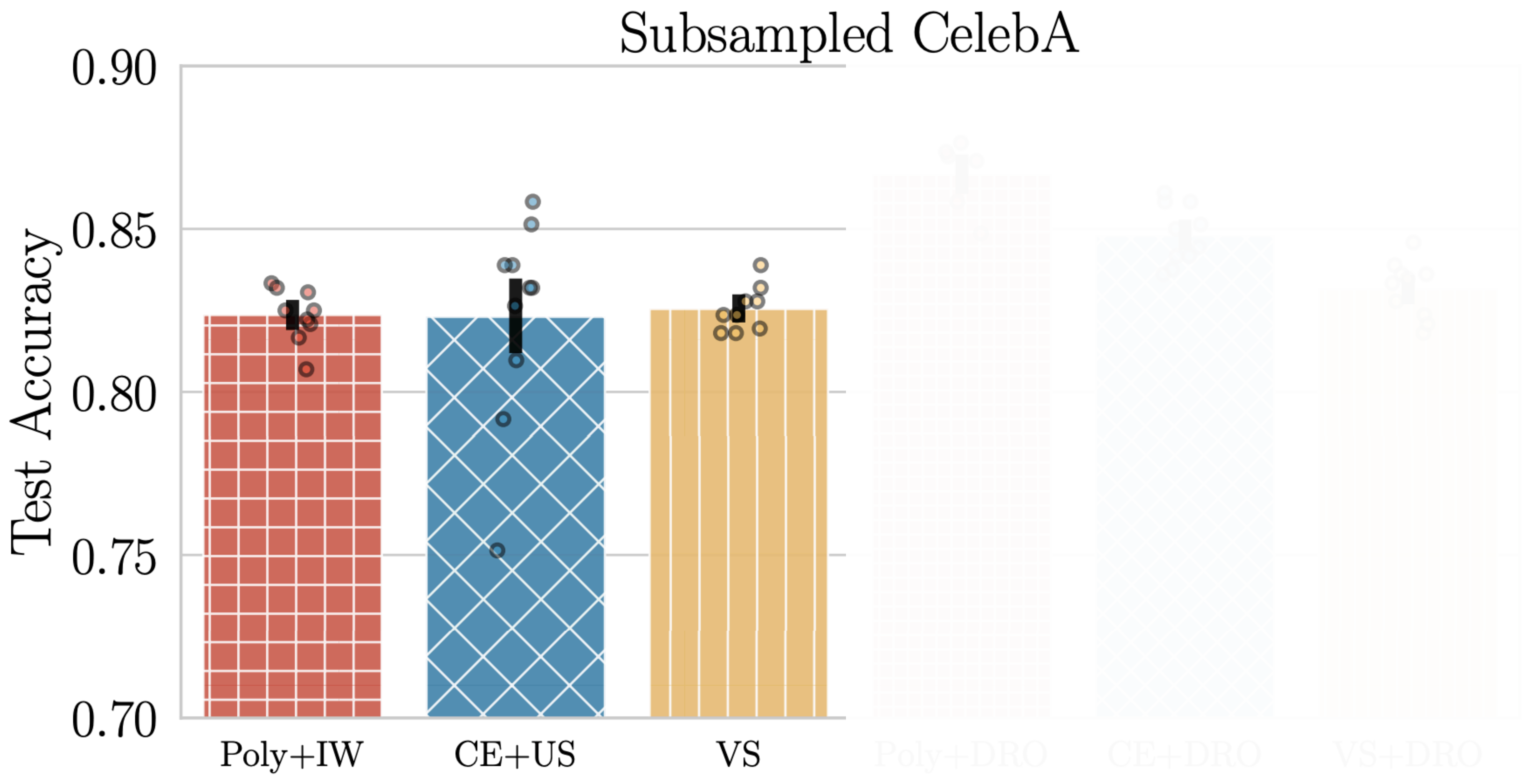
exp. weights

Polynomial Losses + exponentiated weights improve performance for NNs

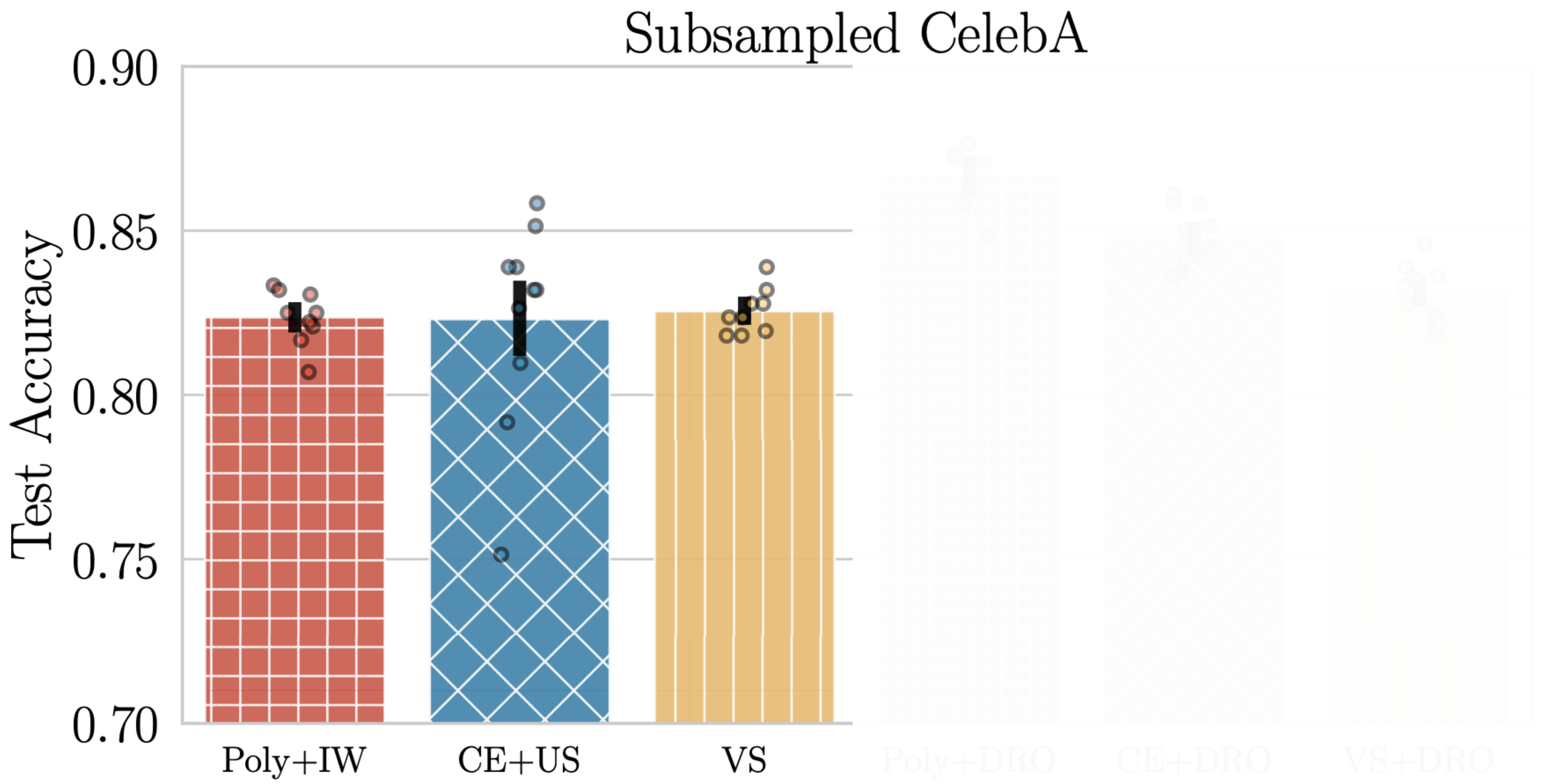
Performance improves even when regularization is used

Comparison with Stronger Reweighting Methods

Comparison with Stronger Reweighting Methods

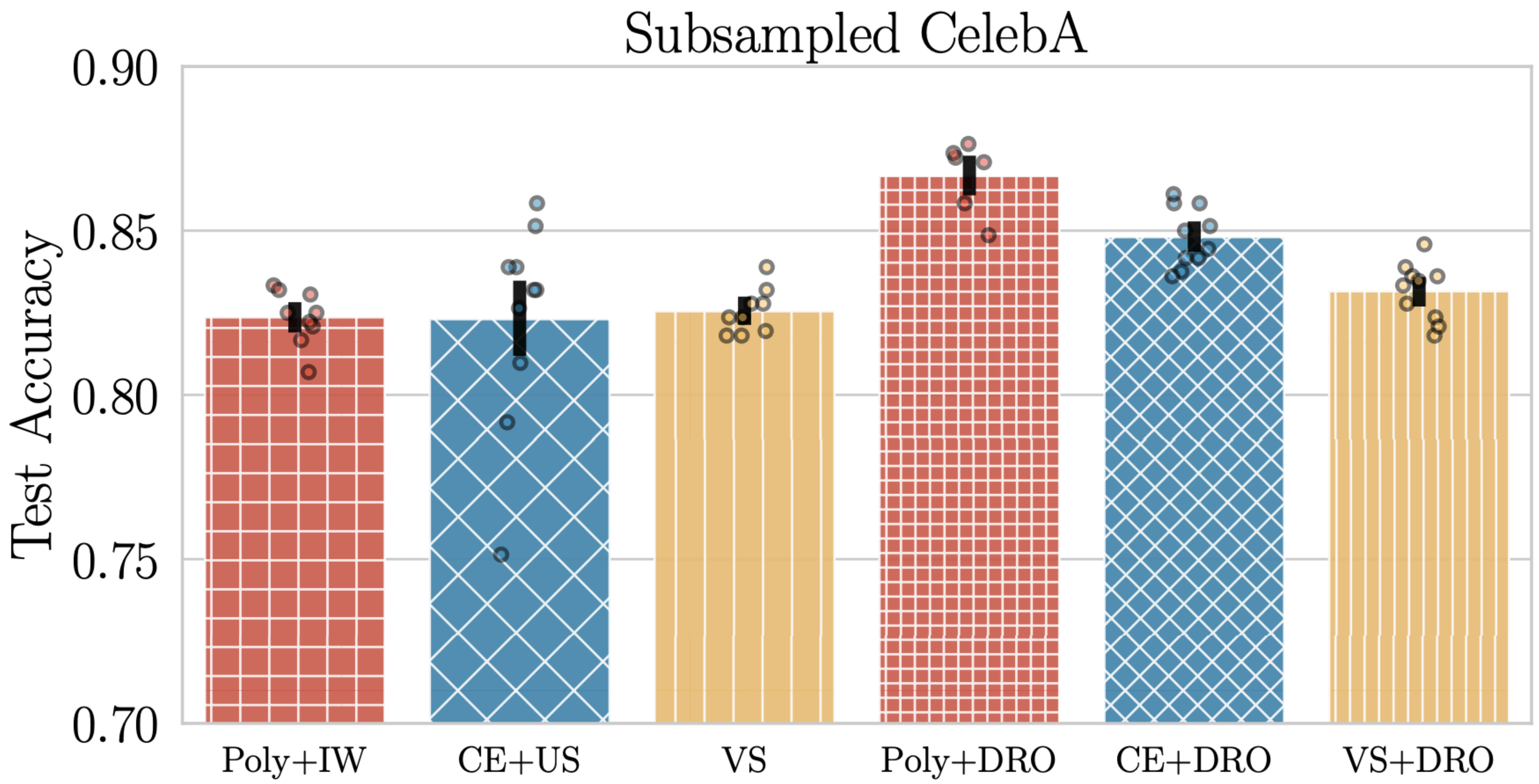


Comparison with Stronger Reweighting Methods



Reweighted poly-loss is competitive current best reweighting methods

Comparison with Stronger Reweighting Methods



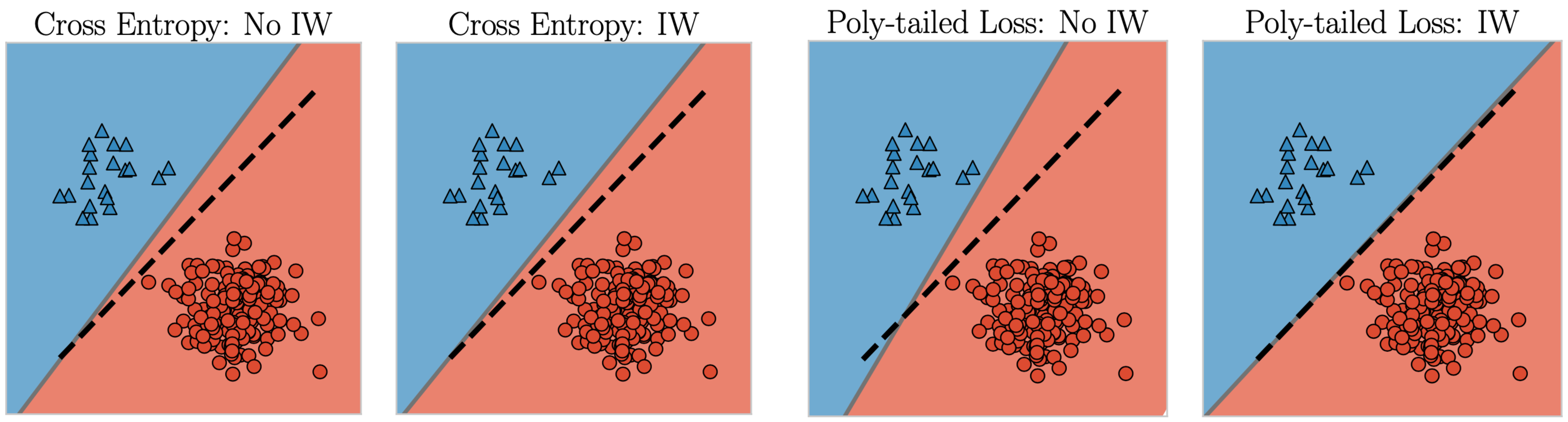
Reweighted poly-loss is competitive current best reweighting methods

Also possible to plug into sophisticated DRO methods and see improvements

Takeaways

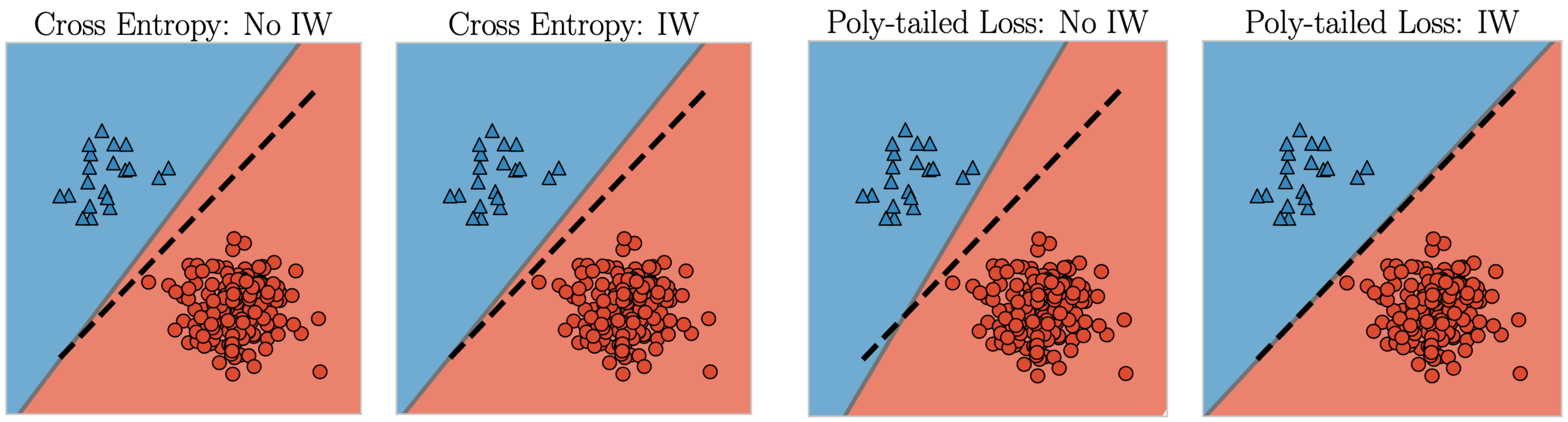
Takeaways

- Robustness interventions behave differently in the interpolation regime

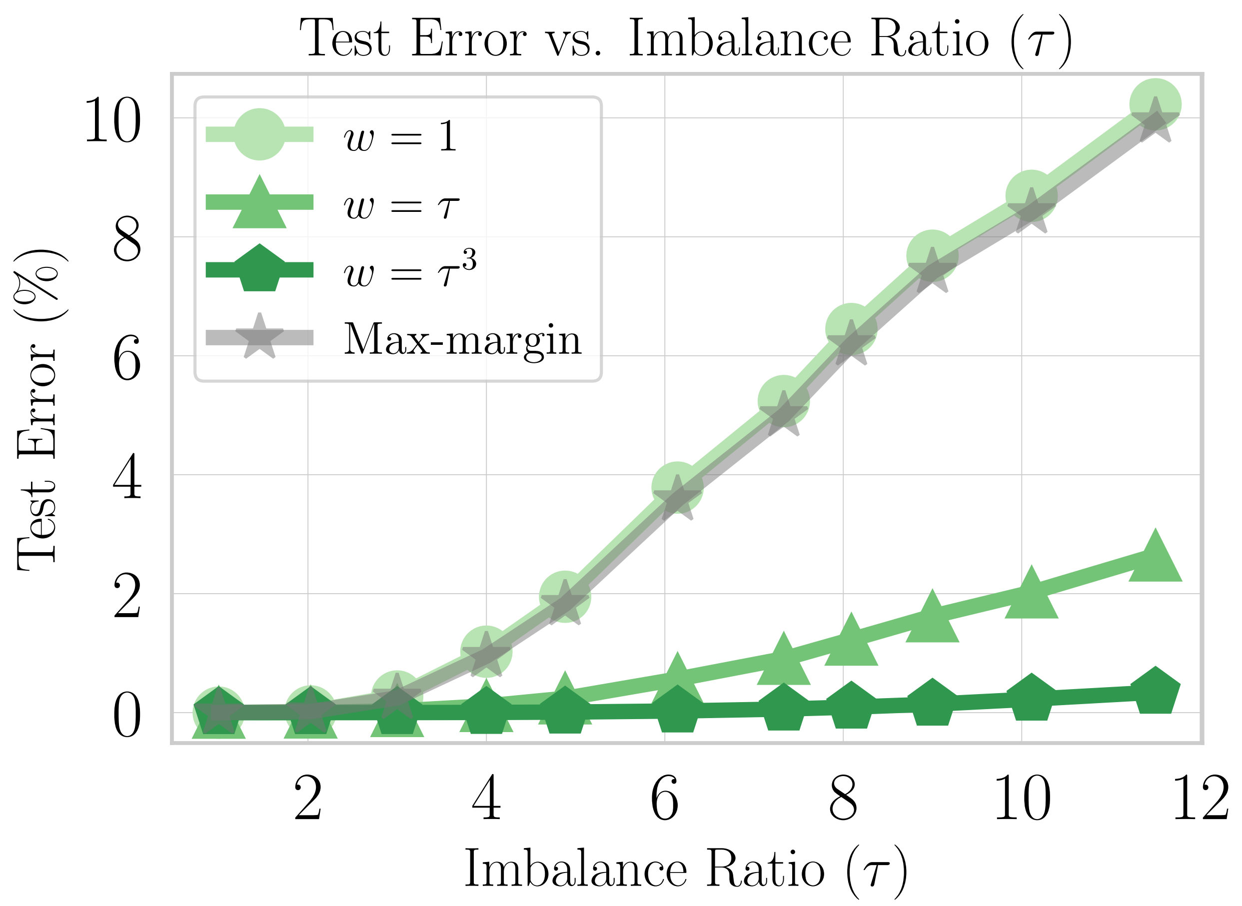


Takeaways

- Robustness interventions behave differently in the interpolation regime



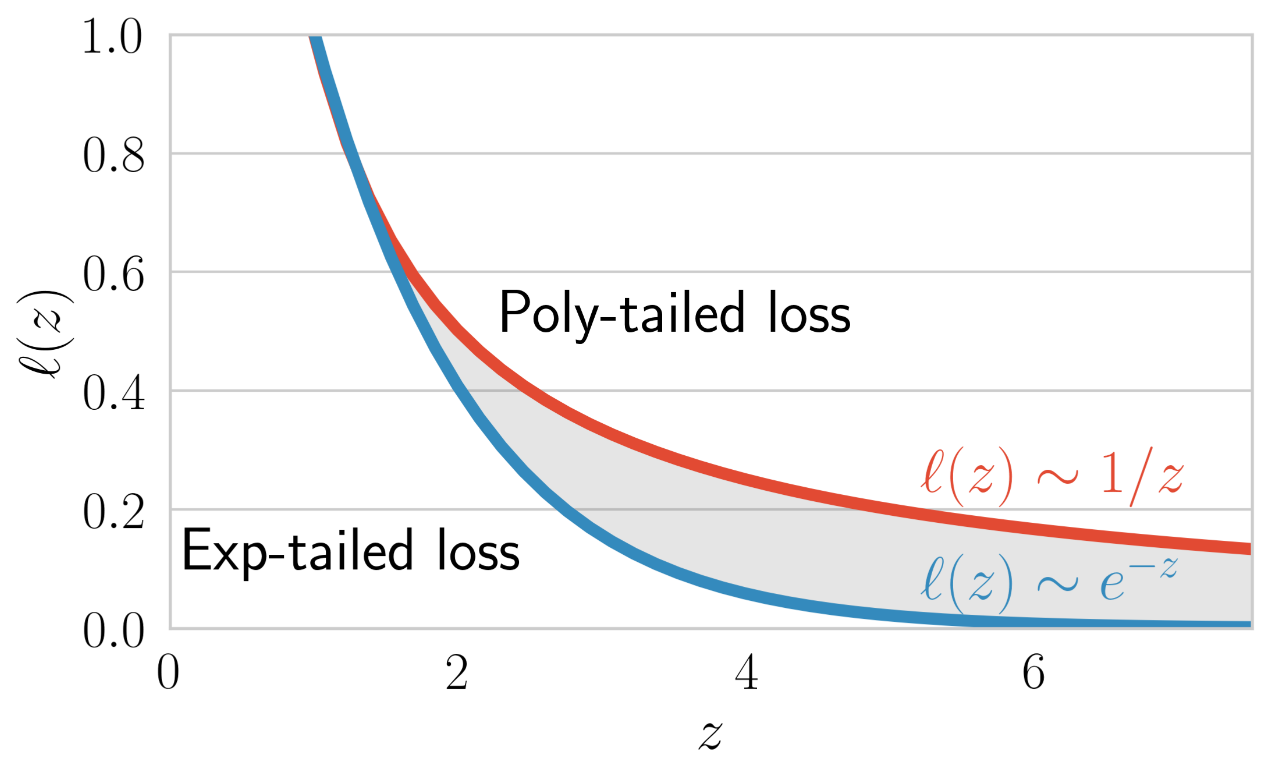
- Careful theoretical analysis leads us to new non-intuitive interventions



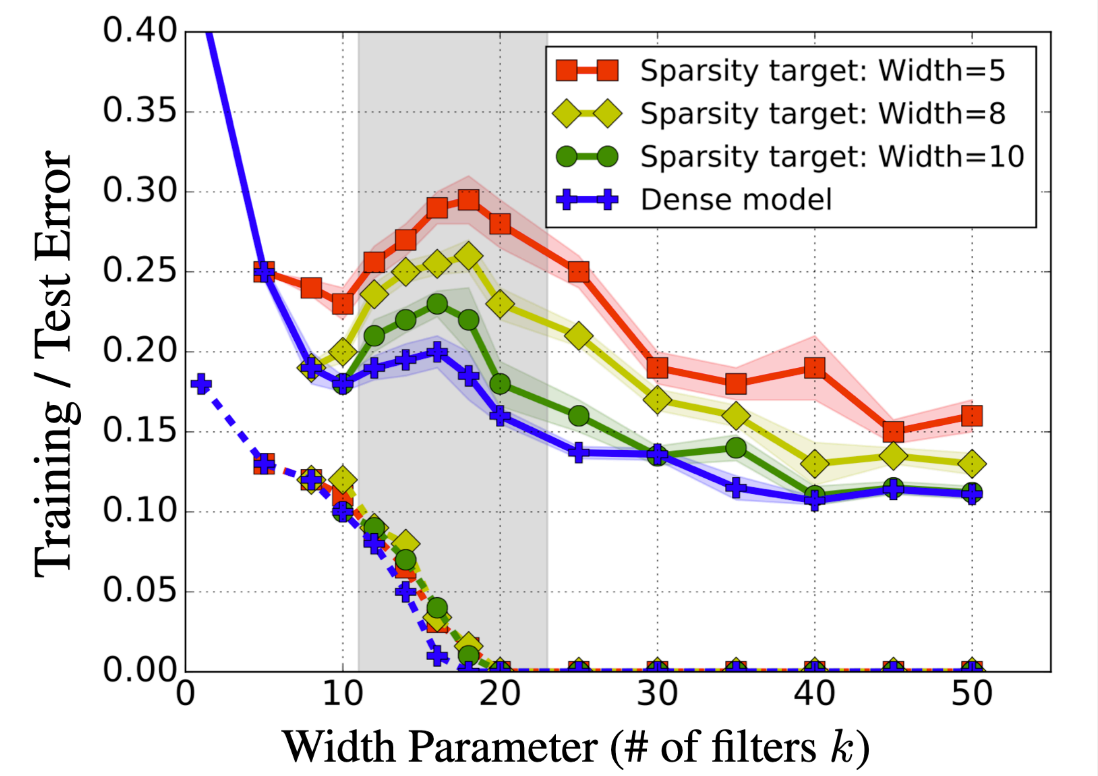
Talk Outline

Study these non-standard settings with linear models

Interpolation under Distribution Shift



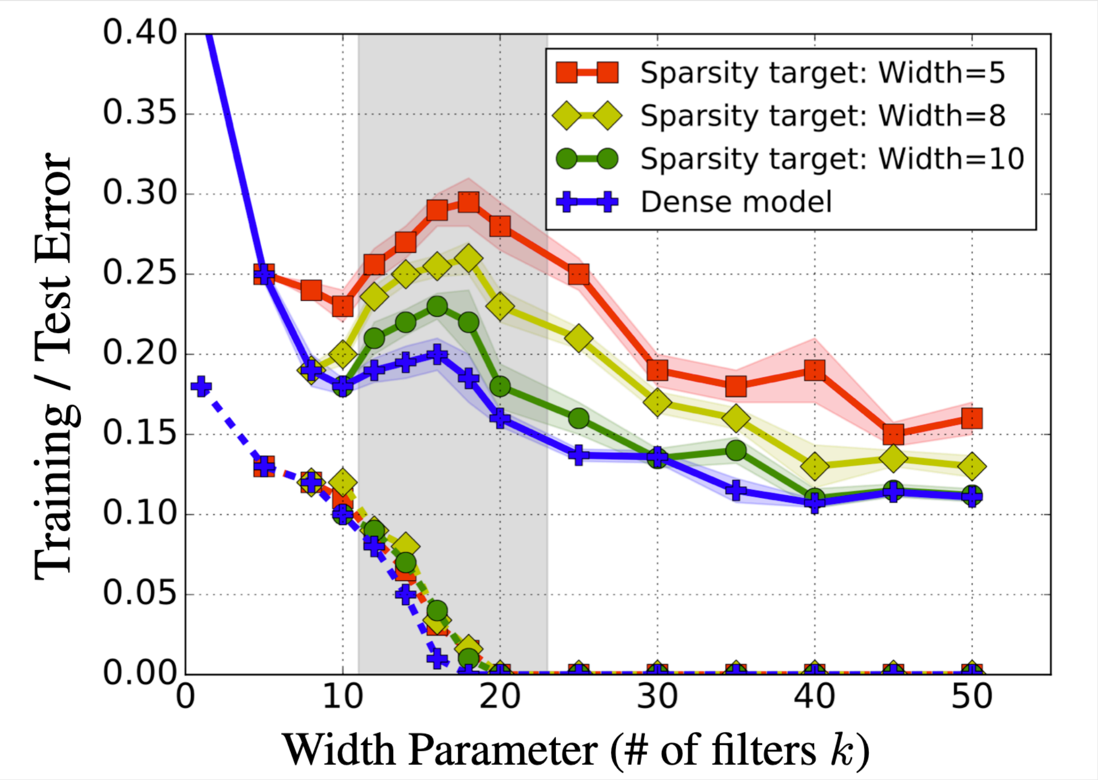
Sparsity and Interpolation



Talk Outline

Study these non-standard settings with linear models

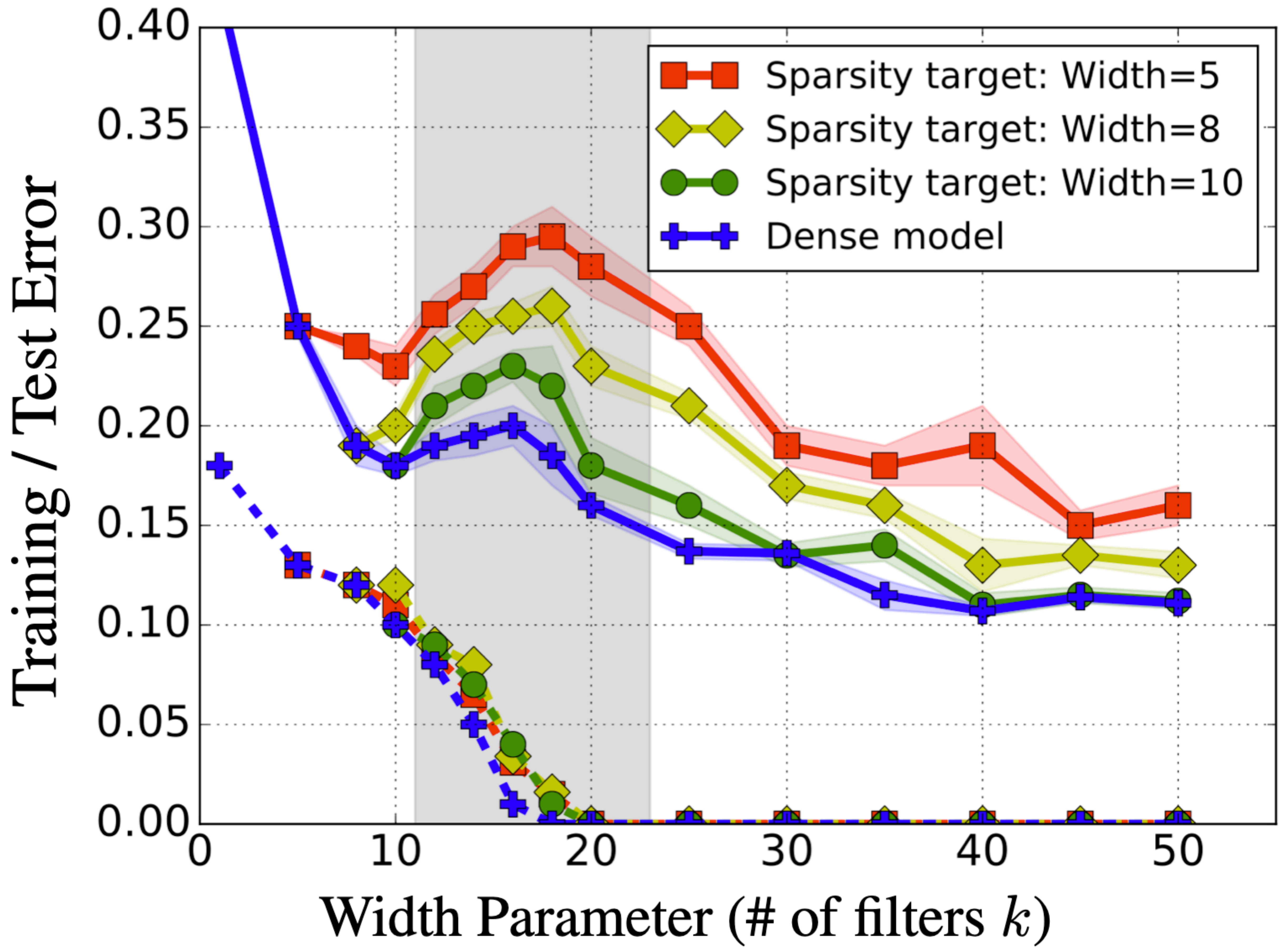
Sparsity and Interpolation



Is sparsity incompatible with interpolation?

ResNet20 trained on CIFAR10

(Chan et al. 2021)



Sparsity seems to hurt the test error

Sparsity in Linear Regression

Sparsity in Linear Regression

Given n datapoints, $(x_1, y_1), \ldots, (x_n, y_n)$

Sparsity in Linear Regression

Given n datapoints, $(x_1, y_1), \ldots, (x_n, y_n)$

1. $x_i \in \mathbb{R}^d$, with $d > n$

Sparsity in Linear Regression

Given n datapoints, $(x_1, y_1), \ldots, (x_n, y_n)$

1. $x_i \in \mathbb{R}^d$, with $d > n$

2. $y_i = \langle x_i, \theta^\star \rangle + \xi_i$, where $\theta^\star$ is k-sparse

Sparsity in Linear Regression

Given n datapoints, $(x_1, y_1), \ldots, (x_n, y_n)$

1. $x_i \in \mathbb{R}^d$, with $d > n$

2. $y_i = \langle x_i, \theta^\star \rangle + \xi_i$, where $\theta^\star$ is k-sparse

Is there an interpolant that leverages this underlying sparsity?

Sparsity in Linear Regression

Q: How does the excess risk of a sparse interpolator behave?

1. $x_i \in \mathbb{R}^d$, with $d > n$

2. $y_i = \langle x_i, \theta^\star \rangle + \xi_i$, where $\theta^\star$ is k-sparse

Is there an interpolant that leverages this underlying sparsity?

Sparsity in Linear Regression

Q: How does the excess risk of a sparse interpolator behave?

1. $x_i \in \mathbb{R}^d$, with $d > n$

Example: the minimum ℓ_1-norm interpolant is defined as

2. $y_i = \langle x_i, \theta^\star \rangle + \xi_i$, where $\theta^\star$ is k-sparse

Is there an interpolant that leverages this underlying sparsity?

Sparsity in Linear Regression

Q: How does the excess risk of a sparse interpolator behave?

Example: the minimum ℓ_1-norm interpolant is defined as

$$\theta_{\ell_1} \in \arg\min_{\theta \in \mathbb{R}^d} ||\theta||_1, \text{ such that } \underset{\mathbb{R}^n}{\mathbf{y}} = \underset{\mathbb{R}^{n\times d}}{X}\underset{\mathbb{R}^d}{\theta}$$

Sparsity in Linear Regression

Q: How does the excess risk of a sparse interpolator behave?

Example: the minimum ℓ_1-norm interpolant is defined as

$$\theta_{\ell_1} \in \arg\min_{\theta \in \mathbb{R}^d} ||\theta||_1, \text{ such that } \underset{\mathbb{R}^n}{\mathbf{y}} = \underset{\mathbb{R}^{n\times d}}{X}\underset{\mathbb{R}^d}{\theta}$$

θ_{ℓ_1} (Basis Pursuit) is known to promote sparsity

Is there an interpolant that leverages this underlying sparsity?

Sparsity in Linear Regression

Q: How does the excess risk of a sparse interpolator behave?

Example: the minimum ℓ_1-norm interpolant is defined as

$$\theta_{\ell_1} \in \arg\min_{\theta \in \mathbb{R}^d} ||\theta||_1, \text{ such that } \underset{\mathbb{R}^n}{\mathbf{y}} = \underset{\mathbb{R}^{n\times d}}{X}\,\underset{\mathbb{R}^d}{\theta}$$

θ_{ℓ_1} (Basis Pursuit) is known to promote sparsity

Q: Does a spase interpolant outperform dense interpolants (min ℓ_2-norm)?

Sparsity in Linear Regression

Q: How does the excess risk of a sparse interpolator behave?

Example: the minimum ℓ_1-norm interpolant is defined as

$$\theta_{\ell_1} \in \arg\min_{\theta \in \mathbb{R}^d} \|\theta\|_1, \text{ such that } \underset{\mathbb{R}^n}{\mathbf{y}} = \underset{\mathbb{R}^{n\times d}}{X}\underset{\mathbb{R}^d}{\theta}$$

θ_{ℓ_1} (Basis Pursuit) is known to promote sparsity

Q: Does a spase interpolant outperform dense interpolants (min ℓ_2-norm)?

We show that sparsity is *incompatible* with interpolation by a lower bound

Construction for the Lower bound

Given n datapoints, $(x_1, y_1), \ldots, (x_n, y_n)$, where $x_i \in \mathbb{R}^d$ and $y_i = \langle x_i, \theta^\star \rangle + \xi_i$

Under the following assumptions:

Construction for the Lower bound

Given n datapoints, $(x_1, y_1), \ldots, (x_n, y_n)$, where $x_i \in \mathbb{R}^d$ and $y_i = \langle x_i, \theta^\star \rangle + \xi_i$

Under the following assumptions:

1. The coordinates of x drawn from $\mathbf{N}(0,\Sigma)$

Construction for the Lower bound

Given n datapoints, $(x_1, y_1), \ldots, (x_n, y_n)$, where $x_i \in \mathbb{R}^d$ and $y_i = \langle x_i, \theta^\star \rangle + \xi_i$

Under the following assumptions:

1. The coordinates of x drawn from $\mathbf{N}(0,\Sigma)$

(k, ε) Model

$$\lambda_1 = \ldots = \lambda_k = 1$$
$$\lambda_{k+1} = \ldots = \lambda_d = \epsilon$$

$$\Sigma = \begin{bmatrix} I_{k\times k} & 0 \\ 0 & \varepsilon \cdot I_{d-k\times d-k} \end{bmatrix}$$

Construction for the Lower bound

Given n datapoints, $(x_1, y_1), \ldots, (x_n, y_n)$, where $x_i \in \mathbb{R}^d$ and $y_i = \langle x_i, \theta^\star \rangle + \xi_i$

Under the following assumptions:

1. The coordinates of x drawn from $\mathbf{N}(0,\Sigma)$
2. The noise drawn independently $\xi \sim \mathbf{N}(0,\sigma^2)$

(k, ε) Model

$$\lambda_1 = \ldots = \lambda_k = 1$$
$$\lambda_{k+1} = \ldots = \lambda_d = \epsilon$$

$$\Sigma = \begin{bmatrix} I_{k\times k} & 0 \\ 0 & \varepsilon \cdot I_{d-k\times d-k} \end{bmatrix}$$

Construction for the Lower bound

Given n datapoints, $(x_1, y_1), \ldots, (x_n, y_n)$, where $x_i \in \mathbb{R}^d$ and $y_i = \langle x_i, \theta^\star \rangle + \xi_i$

Under the following assumptions:

1. The coordinates of x drawn from $\mathbf{N}(0,\Sigma)$
2. The noise drawn independently $\xi \sim \mathbf{N}(0,\sigma^2)$
3. The true model $\theta^\star$ is k-sparse

(k, ε) Model

$$\lambda_1 = \ldots = \lambda_k = 1$$
$$\lambda_{k+1} = \ldots = \lambda_d = \epsilon$$

$$\Sigma = \begin{bmatrix} I_{k\times k} & 0 \\ 0 & \varepsilon \cdot I_{d-k\times d-k} \end{bmatrix}$$

Lower bound for *Any* Sparse Linear Interpolator

Lower bound for *Any* Sparse Linear Interpolator

Theorem: For any $\delta \in (0,1/2)$ if
$\sigma \gtrsim \|\theta^\star\|, d \gtrsim n$ and $n \gtrsim \log^2(1/\delta) + k^{1+c}$

Lower bound for *Any* Sparse Linear Interpolator

Theorem: For any $\delta \in (0,1/2)$ if
$\sigma \gtrsim \|\theta^\star\|$, $d \gtrsim n$ and $n \gtrsim \log^2(1/\delta) + k^{1+c}$
then with probability $1-\delta$, *any s-sparse interpolator* θ_s satisfies

Lower bound for *Any* Sparse Linear Interpolator

Theorem: For any $\delta \in (0,1/2)$ if
$\sigma \gtrsim \|\theta^\star\|, d \gtrsim n$ and $n \gtrsim \log^2(1/\delta) + k^{1+c}$
then with probability $1 - \delta$, *any s-sparse interpolator* θ_s satisfies

$$\mathrm{R}(\theta_s) := \|\theta_s - \theta^\star\|_\Sigma^2 \gtrsim \frac{\sigma^2 n}{s \log^2(d/s)}$$

Lower bound for *Any* Sparse Linear Interpolator

Theorem: For any $\delta \in (0,1/2)$ if
$\sigma \gtrsim \|\theta^\star\|, d \gtrsim n$ and $n \gtrsim \log^2(1/\delta) + k^{1+c}$
then with probability $1 - \delta$, *any s-sparse interpolator* θ_s satisfies

$$\mathsf{R}(\theta_s) := \|\theta_s - \theta^\star\|_\Sigma^2 \gtrsim \frac{\sigma^2 n}{s \log^2(d/s)}$$

Lower bound for *Any* Sparse Linear Interpolator

Theorem: For any $\delta \in (0,1/2)$ if
$\sigma \gtrsim \|\theta^\star\|$, $d \gtrsim n$ and $n \gtrsim \log^2(1/\delta) + k^{1+c}$
then with probability $1-\delta$, *any s-sparse interpolator* θ_s satisfies

$$\mathsf{R}(\theta_s) := \|\theta_s - \theta^\star\|_\Sigma^2 \gtrsim \frac{\sigma^2 n}{s \log^2(d/s)}$$

Lower bound for *Any* Sparse Linear Interpolator

Theorem: For any $\delta \in (0,1/2)$ if
$\sigma \gtrsim \|\theta^\star\|, d \gtrsim n$ and $n \gtrsim \log^2(1/\delta) + k^{1+c}$
then with probability $1-\delta$, *any* s*-sparse interpolator* θ_s satisfies

$$\mathsf{R}(\theta_s) := \|\theta_s - \theta^\star\|_\Sigma^2 \gtrsim \frac{\sigma^2 n}{s \log^2(d/s)}$$

(Similar bound in the isotropic case by Muthukumar et al. 2020)

Risk is larger for Sparser Models

$$\mathrm{R}(\theta) \gtrsim \frac{\sigma^2 n}{s \log^2(d/s)}$$

Risk is larger for Sparser Models

$$\mathsf{R}(\theta) \gtrsim \frac{\sigma^2 n}{s \log^2(d/s)}$$

Sparsity Level (s)

$$k \quad < \quad n \quad < \quad d$$

Risk is larger for Sparser Models

$$\mathrm{R}(\theta) \gtrsim \frac{\sigma^2 n}{s \log^2(d/s)}$$

Sparsity Level (s)

$$k \quad < \quad n \quad < \quad d$$

$$\frac{\sigma^2 n}{k \log^2(d/n)} \quad > \quad \frac{\sigma^2}{\log^2(d/n)} \quad > \quad \frac{\sigma^2 n}{d}$$

$R(\theta)$

What about the Min ℓ_1-norm Interpolant?

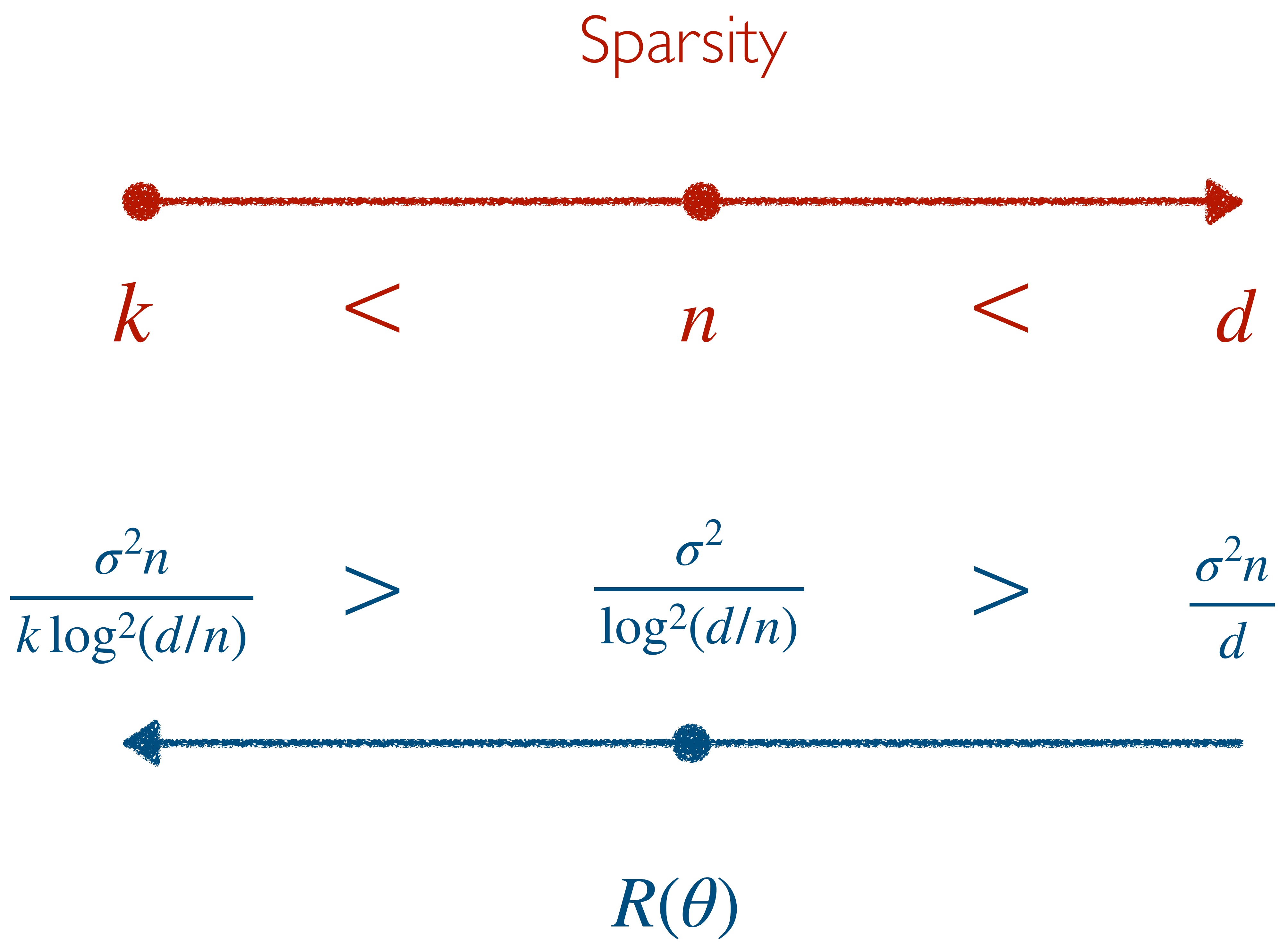
Sparsity

$$k \quad < \quad n \quad < \quad d$$

$$\frac{\sigma^2 n}{k \log^2(d/n)} \quad > \quad \frac{\sigma^2}{\log^2(d/n)} \quad > \quad \frac{\sigma^2 n}{d}$$

$R(\theta)$

What about the Min ℓ_1-norm Interpolant?



Basis pursuit outputs n-sparse model a.s.

What about the Min ℓ_1-norm Interpolant?

Sparsity

$k \quad < \quad n \quad < \quad d$

$\dfrac{\sigma^2 n}{k \log^2(d/n)} \quad > \quad \dfrac{\sigma^2}{\log^2(d/n)} \quad > \quad \dfrac{\sigma^2 n}{d}$

$R(\theta)$

Basis pursuit outputs n-sparse model a.s.

$$\mathsf{R}(\theta_{\ell_1}) \gtrsim \frac{\sigma^2}{\log^2(d/n)}$$

$(\theta_{\ell_1} = \arg\min_{\theta \in \mathbb{R}^d} \|\theta\|_1, \text{ s.t. } \mathbf{y} = X\theta)$

What about the Min ℓ_1-norm Interpolant?

Why is this bad?

Basis pursuit outputs n-sparse model a.s.

$$R(\theta_{\ell_1}) \gtrsim \frac{\sigma^2}{\log^2(d/n)}$$

$$(\theta_{\ell_1} = \arg\min_{\theta \in \mathbb{R}^d} \|\theta\|_1, \text{ s.t. } y = X\theta)$$

What about the Min ℓ_1-norm Interpolant?

Why is this bad?

Consider the case where

- $k = \sqrt{n}$
- $d = n^2$
- $\varepsilon = 1/n^2$
- $\sigma^2 = \|\theta^\star\|^2 = 1$

What about the Min ℓ_1-norm Interpolant?

Why is this bad?

Consider the case where

- $k = \sqrt{n}$
- $d = n^2$
- $\varepsilon = 1/n^2$
- $\sigma^2 = \|\theta^\star\|^2 = 1$

$$\boxed{\mathrm{R}(\theta_{\ell_2}) = \mathcal{O}\left(\frac{1}{\sqrt{n}}\right)}$$

(Bartlett et al. 2019) *Dense* Min. ℓ_2-norm (OLS)

What about the Min ℓ_1-norm Interpolant?

Why is this bad?

Consider the case where

- $k = \sqrt{n}$
- $d = n^2$
- $\varepsilon = 1/n^2$
- $\sigma^2 = \|\theta^\star\|^2 = 1$

$$\mathrm{R}(\theta_{\ell_2}) = \mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$$

(Bartlett et al. 2019) *Dense* Min. ℓ_2-norm (OLS)

$$\mathrm{R}(\theta_{\ell_1}) = \Omega\left(\frac{1}{\log^2 n}\right)$$

Sparse Min. ℓ_1-norm (BP)

What about the Min ℓ_1-norm Interpolant?

Why is this bad?

Consider the case where

- $k = \sqrt{n}$
- $d = n^2$
- $\varepsilon = 1/n^2$
- $\sigma^2 = \|\theta^\star\|^2 = 1$

$$\mathrm{R}(\theta_{\ell_2}) = \mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$$

(Bartlett et al. 2019) *Dense* Min. ℓ_2-norm (OLS)

$$\mathrm{R}(\theta_{\ell_1}) = \Omega\left(\frac{1}{\log^2 n}\right)$$

Sparse Min. ℓ_1-norm (BP)

Exponential Slowdown!

What about the Min ℓ_1-norm Interpolant?

Why is this bad?

Consider the case where

- $k = \sqrt{n}$
- $d = n^2$
- $\varepsilon = 1/n^2$
- $\sigma^2 = \|\theta^\star\|^2 = 1$

$$\mathrm{R}(\theta_{\ell_2}) = \mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$$

(Bartlett et al. 2019) *Dense* Min. ℓ_2-norm (OLS)

$$\mathrm{R}(\theta_{\ell_1}) = \Omega\left(\frac{1}{\log^2 n}\right)$$

Sparse Min. ℓ_1-norm (BP)

Exponential Slowdown!

Nearly matching upper bounds (Koehler et al., Wang et al., Li and Wei 2021, Donhauser et al. 2021)

Intuition

Intuition

Energy of the noise scales as $||\mathbf{y}||^2 \geq \sigma^2 n$

Intuition

Energy of the noise scales as $\|\mathbf{y}\|^2 \geq \sigma^2 n$

Dense interpolators like the OLS can spread this over d directions

Intuition

Energy of the noise scales as $\|\mathbf{y}\|^2 \geq \sigma^2 n$

Dense interpolators like the OLS can spread this over d directions

$$\mathbf{y} = X\theta = \begin{bmatrix} x_{11} & \cdots & x_{1k} & x_{1k+1} & \cdots & x_{1d} \\ x_{21} & \cdots & x_{2k} & x_{2k+1} & \cdots & x_{2d} \\ \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ x_{n1} & \cdots & x_{nk} & x_{nk+1} & \cdots & x_{nd} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_d \end{bmatrix}$$

Intuition

Energy of the noise scales as $\|\mathbf{y}\|^2 \geq \sigma^2 n$

Dense interpolators like the OLS can spread this over d directions

$$\mathbf{y} = X\theta = \begin{bmatrix} x_{11} & \cdots & x_{1k} & x_{1k+1} & \cdots & x_{1d} \\ x_{21} & \cdots & x_{2k} & x_{2k+1} & \cdots & x_{2d} \\ \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ x_{n1} & \cdots & x_{nk} & x_{nk+1} & \cdots & x_{nd} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_d \end{bmatrix}$$

Including many *unimportant* directions

Intuition

Energy of the noise scales as $\|\mathbf{y}\|^2 \geq \sigma^2 n$

Dense interpolators like the OLS can spread this over d directions

$$\mathbf{y} = X\theta = \begin{bmatrix} x_{11} & \cdots & x_{1k} & x_{1k+1} & \cdots & x_{1d} \\ x_{21} & \cdots & x_{2k} & x_{2k+1} & \cdots & x_{2d} \\ \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ x_{n1} & \cdots & x_{nk} & x_{nk+1} & \cdots & x_{nd} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_d \end{bmatrix}$$

Including many *unimportant* directions

However, sparse estimators like BP can only spread it over s directions

Summary and Future Directions

Summary and Future Directions

1. Importance Weighting with Interpolators

Summary and Future Directions

1. Importance Weighting with Interpolators

- Robustness interventions behave differently in the interpolation regime

Summary and Future Directions

1. Importance Weighting with Interpolators

- Robustness interventions behave differently in the interpolation regime
- Careful theoretical analysis leads us to new non-intuitive interventions

Summary and Future Directions

1. Importance Weighting with Interpolators

- Robustness interventions behave differently in the interpolation regime
- Careful theoretical analysis leads us to new non-intuitive interventions

2. Sparsity and Interpolation

Summary and Future Directions

1. Importance Weighting with Interpolators
   - Robustness interventions behave differently in the interpolation regime
   - Careful theoretical analysis leads us to new non-intuitive interventions

2. Sparsity and Interpolation
   - Are other properties are aligned/misaligned with generalization?

Summary and Future Directions

1. Importance Weighting with Interpolators

- Robustness interventions behave differently in the interpolation regime
- Careful theoretical analysis leads us to new non-intuitive interventions

2. Sparsity and Interpolation

- Are other properties are aligned/misaligned with generalization?
- Can we analyze NNs and also understand if sparsity is harmful?