“Optimization is all you need!”

The Deep Bootstrap

Rethinking Generalization to Understand Deep Learning

Preetum Nakkiran
Harvard

Behnam Neyshabur
Google

Hanie Sedghi
Google Brain

Appears in ICLR2021: https://arxiv.org/abs/2010.08127

Motivation

Goal: “Understand” why DL methods used in practice work
(small test error / test loss).

Hope: Predict how design choices affect test error.

This Work: *Framework/roadmap* for achieving goal
(for supervised classification)

Setting (briefly)

Setup: Supervised classification.

Distribution $(x, y) \sim D$

Want: classifier $f(x)$ with small *test error* : $\Pr_{x,y \sim D}[f(x) \neq y]$

Do: SGD on NN to minimize *train error*

Our Framework (high-level)

Classical Framework: Finite train set.

"Good models are those with small generalization gap"

Our Framework: Models trained on finite train set ≈ infinite train set

*"Good models are those which **optimize quickly**, on infinite data"*

Our Framework

Main Idea: compare Real World vs. Ideal World

Fix distribution D, architecture $\mathcal{F}$, num samples n.
Then, for all steps $t \in \mathbb{N}$ define:

Real World(n, t)

SGD on empirical loss

(Train Error $\leq$ Test Error)

$f_t \leftarrow \mathrm{Train}_{\mathcal{F},D}(n, t)$

Test Error

$\approx$

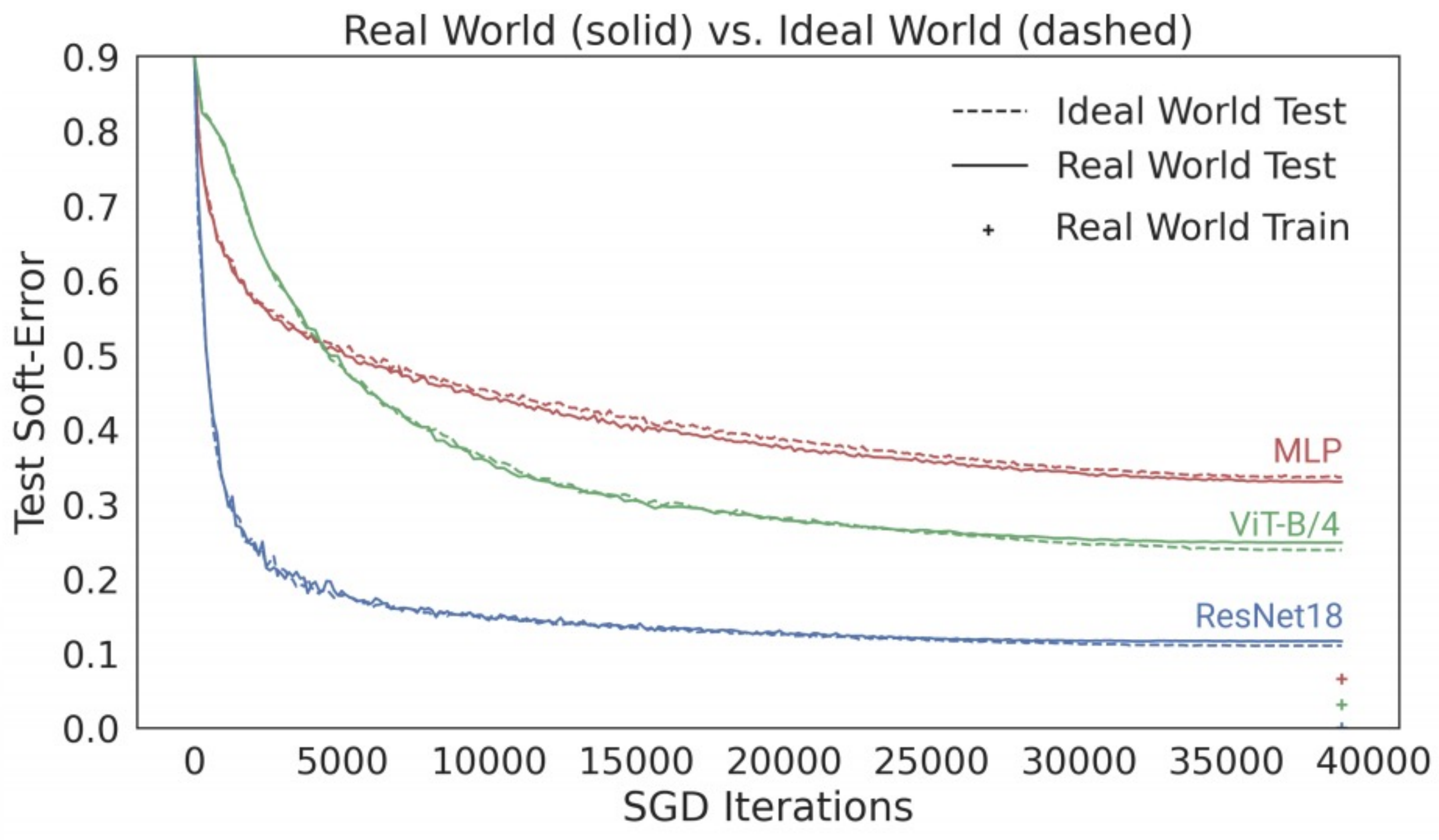
Ideal World(t)

SGD on population loss

(Train Error $\equiv$ Test Error)

$f_t^{\mathrm{iid}} \leftarrow \mathrm{Train}_{\mathcal{F},D}(\infty, t)$

Example



Models which ***optimize faster*** *in Ideal World,* ***generalize better*** *in Real World*

Real World: 50K samples, 100 epochs. **Ideal World:** 5M samples, 1 epoch.

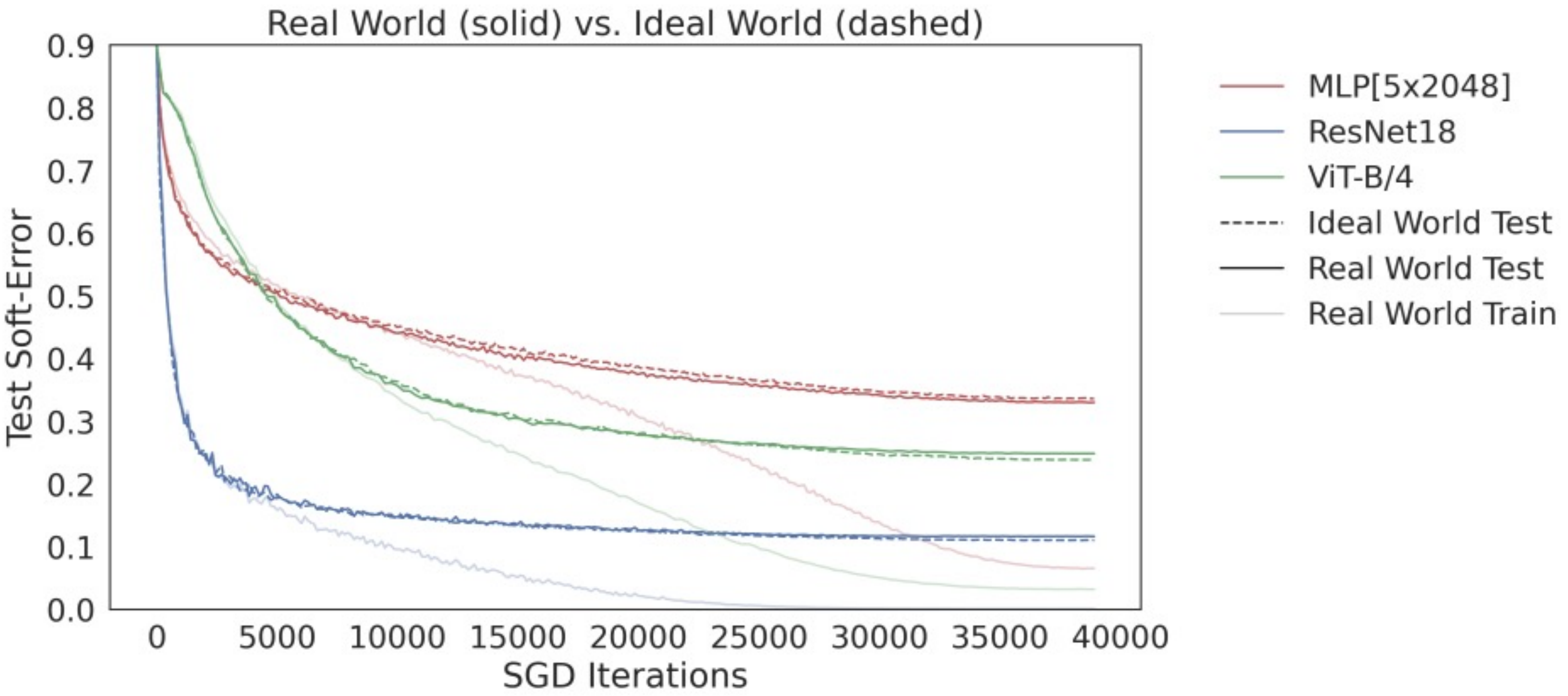


Figure 8: The corresponding train soft-errors for Figure 1.

(More) Precise Claim

SGD on deep nets produces similar models whether trained on ***re-used samples*** *(Real) or* ***fresh samples*** *(Ideal)*

…as measured by Test SoftError

…for as long as the Real World optimizer is still moving (e.g. TrainError $\geq 1\%$*)*

(More) Precise Claim

New decomposition: $$\mathrm{TestError}(f_t) = \underbrace{\mathrm{TestError}(f_t^{\mathrm{iid}})}_{\text{A: Online Learning}} + \underbrace{[\mathrm{TestError}(f_t) - \mathrm{TestError}(f_t^{\mathrm{iid}})]}_{\text{B: Bootstrap error}}$$

Define "bootstrap error" ϵ as (Real – Ideal): $\cdot\, \varepsilon(n, \mathcal{D}, \mathcal{F}, t)$

<u>Main Claim</u>: *Bootstrap error $\epsilon(n, \mathcal{D}, \mathcal{F}, t)$ is small for realistic $(n, \mathcal{D}, \mathcal{F})$, and all $t \leq T_N$.*

Where "stopping time" T_N := time when Real World reaches TrainError $\leq 1\%$.

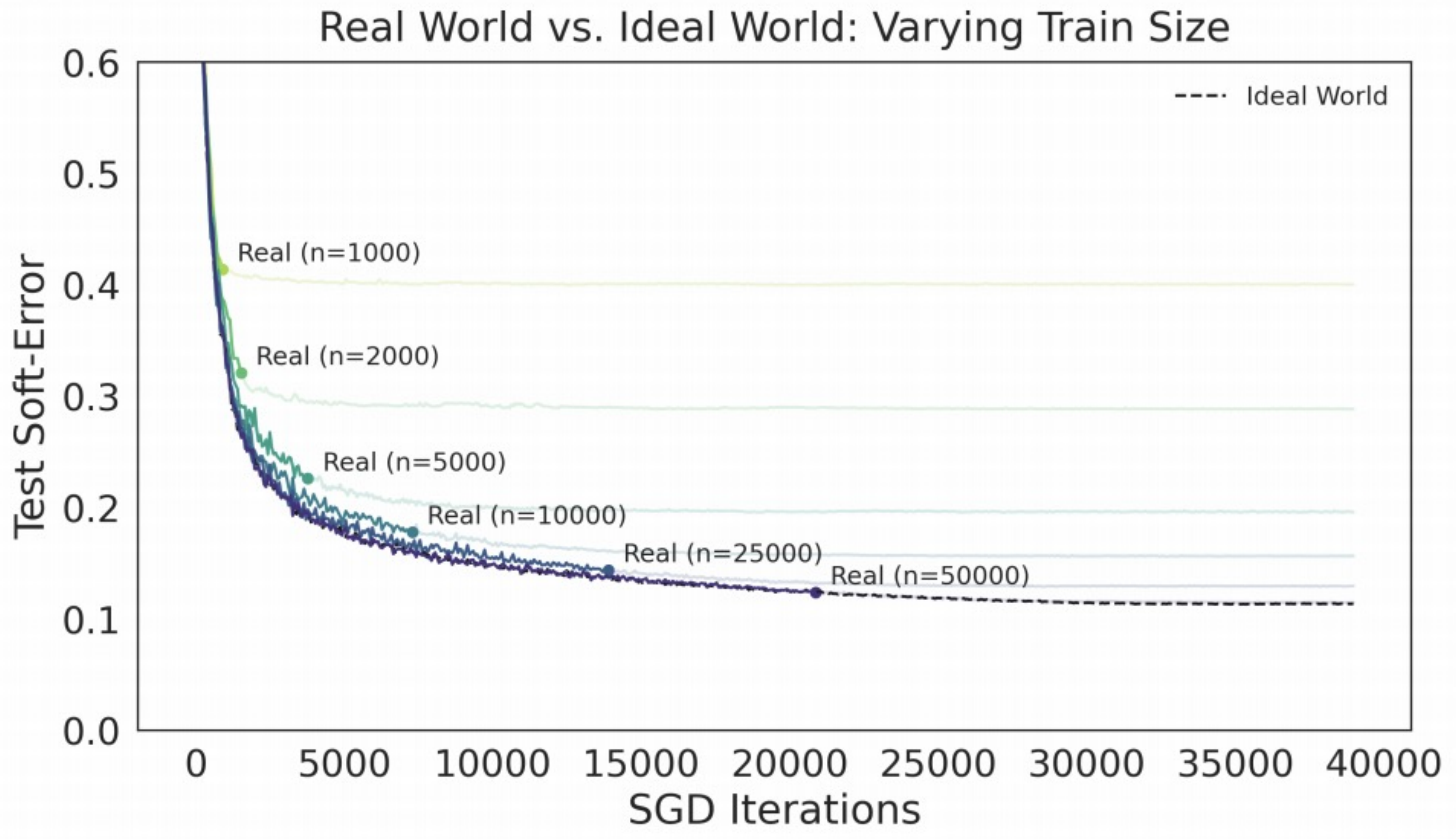
"Deep Bootstrap"

$$\text{RealWorld}(N, T = \infty) \approx \text{RealWorld}(N, T_N) \approx_\epsilon \text{RealWorld}(\infty, T_N)$$

Practice: Real World
(trained as long as possible)

Real World
(stopped at T_N : when Train Error $\approx 1\%$)

Ideal World
(stopped at T_N)



Real World vs. Ideal World: Varying Train Size
Ideal World
Real (n=1000)
Real (n=2000)
Real (n=5000)
Real (n=10000)
Real (n=25000)
Real (n=50000)
Test Soft-Error
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
5000
10000
15000
20000
25000
30000
35000
40000
SGD Iterations

Learning curves:

$L(n)$: Loss on **n** samples (Real-world, trained to convergence)

$T(n)$: Time to converge on **n** samples (Real world SGD steps)

$\tilde{L}(t)$: Loss after **t** online SGD steps (Ideal World)

Then:

$$L(n) \approx \tilde{L}(T(n))$$

Significance

$$\mathrm{TestError}(f_t) = \underbrace{\mathrm{TestError}(f_t^{\mathrm{iid}})}_{\text{A: Online Learning}} + \underbrace{[\mathrm{TestError}(f_t) - \mathrm{TestError}(f_t^{\mathrm{iid}})]}_{\text{B: Bootstrap error}}$$

To understand generalization, sufficient to understand:

1. Online optimization: how fast Ideal World learns.
   [long history, but not in DL]
2. Empirical optimization: how fast Real World convergences (T_N)
   [recent progress: Arora, Allen-Zhu,…]
3. Bootstrap Error: |Real - Ideal|
   [long history in stats, but not in DL]

Assume/prove/believe bootstrap error small ⇒
generalization reduced to **optimization!**

(Towards) Practical Guidance

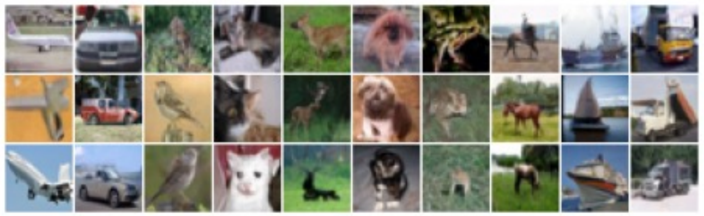
Deep Bootstrap: “Real World ≈ *Ideal World as long as the Real World hasn’t converged”*

Thus, good training procedures:

1. **Optimize quickly** on infinite samples
   (high-capacity models, skip-connections, BN, …)
2. **Don’t optimize too** quickly on finite samples
   (regularization, data-aug,…)

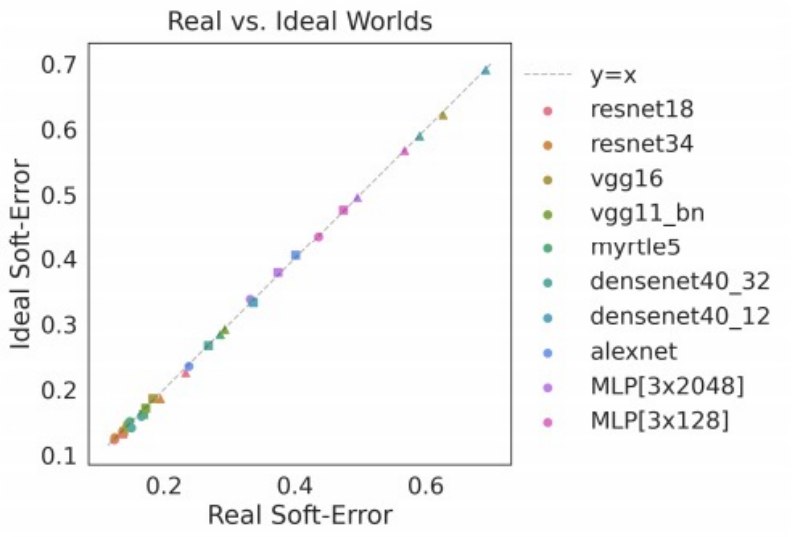
Validation: Summary of Experiments

- **CIFAR-5m:** 5-million synthetic samples from a generative model trained on CIFAR-10



Samples from CIFAR-5m

- **ImageNet-DogBird:** 155K images by collapsing ImageNet catagories. Binary task.

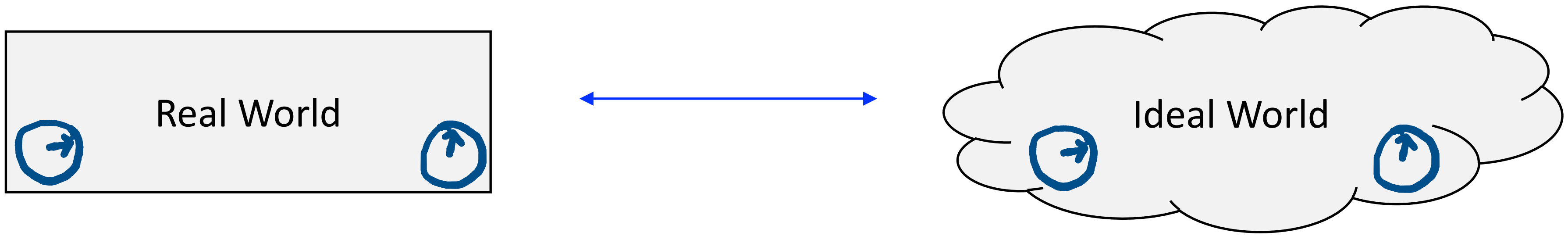


(a) Standard architectures.

Figure 2: **Real vs Ideal World: CIFAR-5m.** SGD w
0.1 (●), 0.01 (■), 0.001 (▲). (b): Random architecture

- **Varying settings**: {archs, opt, LR,...} convnets, ResNets, MLPs, Image-GPT, Vision-Transformer

Implications:
Deep Learning through the Bootstrap Lens



Effect of Pretraining

Pretrained models generalize better (Real)
“because” they optimize faster (Ideal)

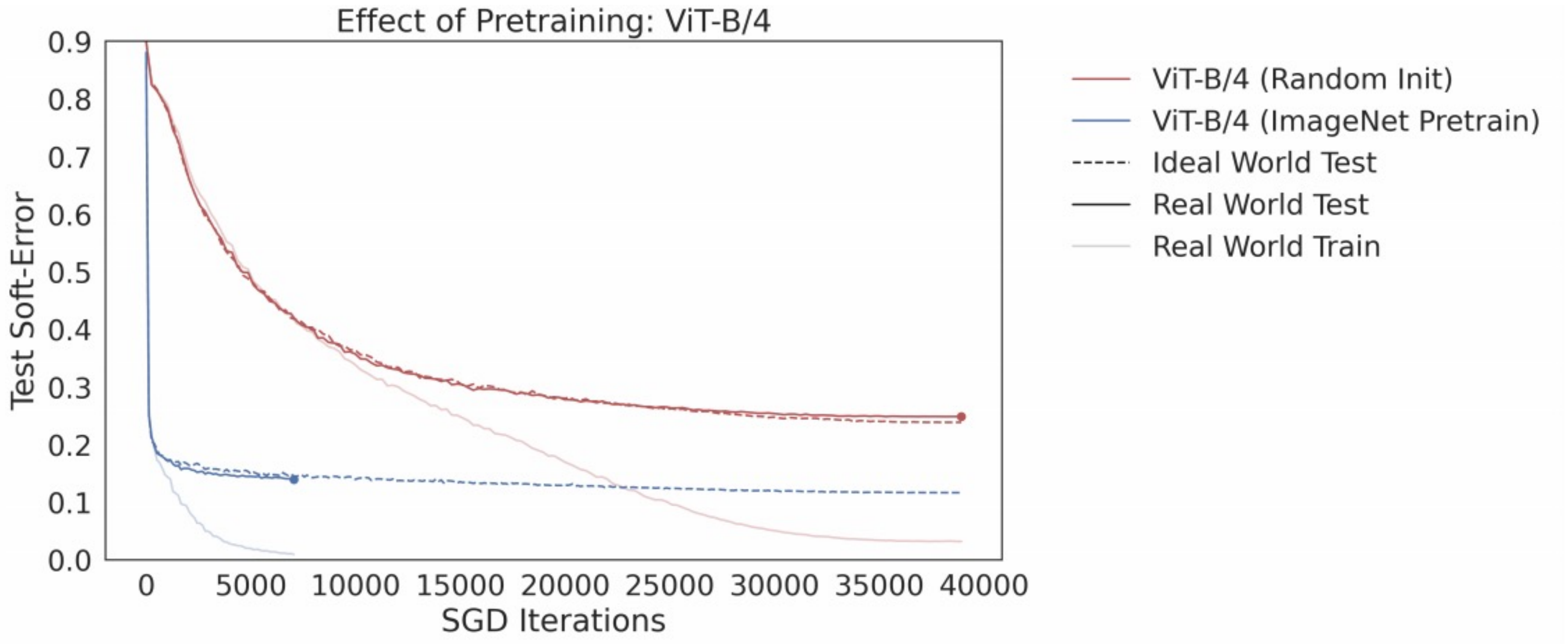
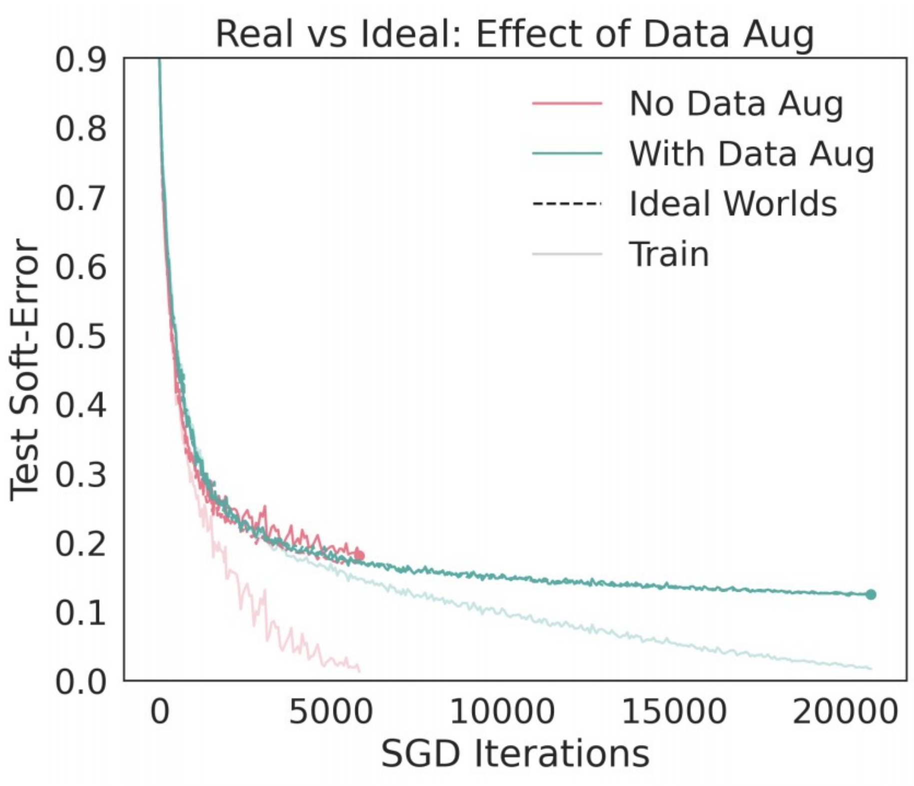


Figure 13: Real vs. Ideal Worlds for Vision Transformer on CIFAR-5m, with and w/o pretraining.

Effect of Data Aug

Data-aug in the Ideal World =
Augment each sample once



3 potential effects:

1. Ideal World Optimization Speed
2. **Real World Convergence Speed**
3. Bootstrap Gap

Good data-augs:

- Don't hurt learning in Ideal World
- Decelerate optimization in Real World (train for longer)

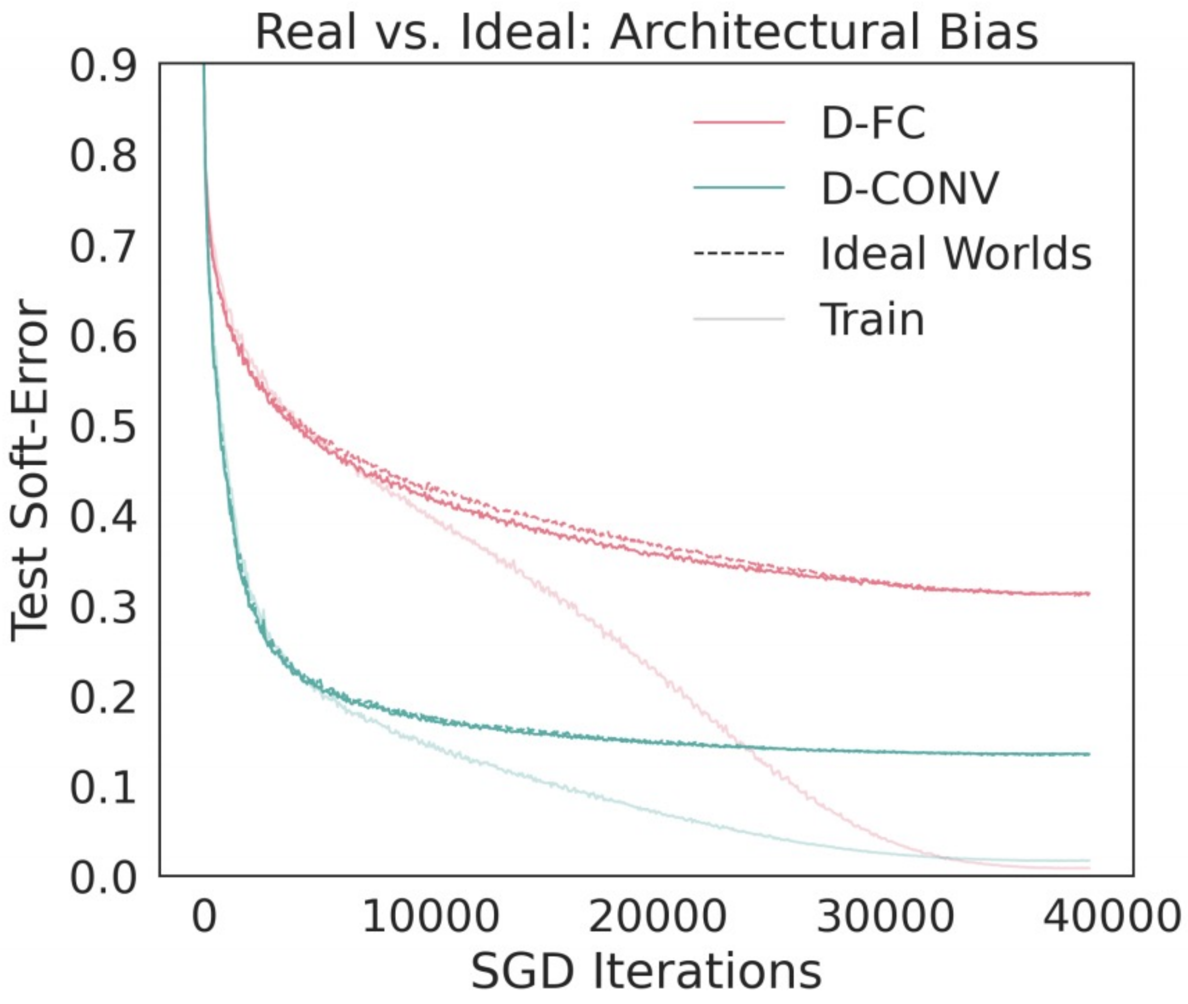
Implicit Bias → Explicit Optimization

Two archs from [Neyshabur 2020]:
D-CONV (convnet) ⊂ D-FC (mlp)

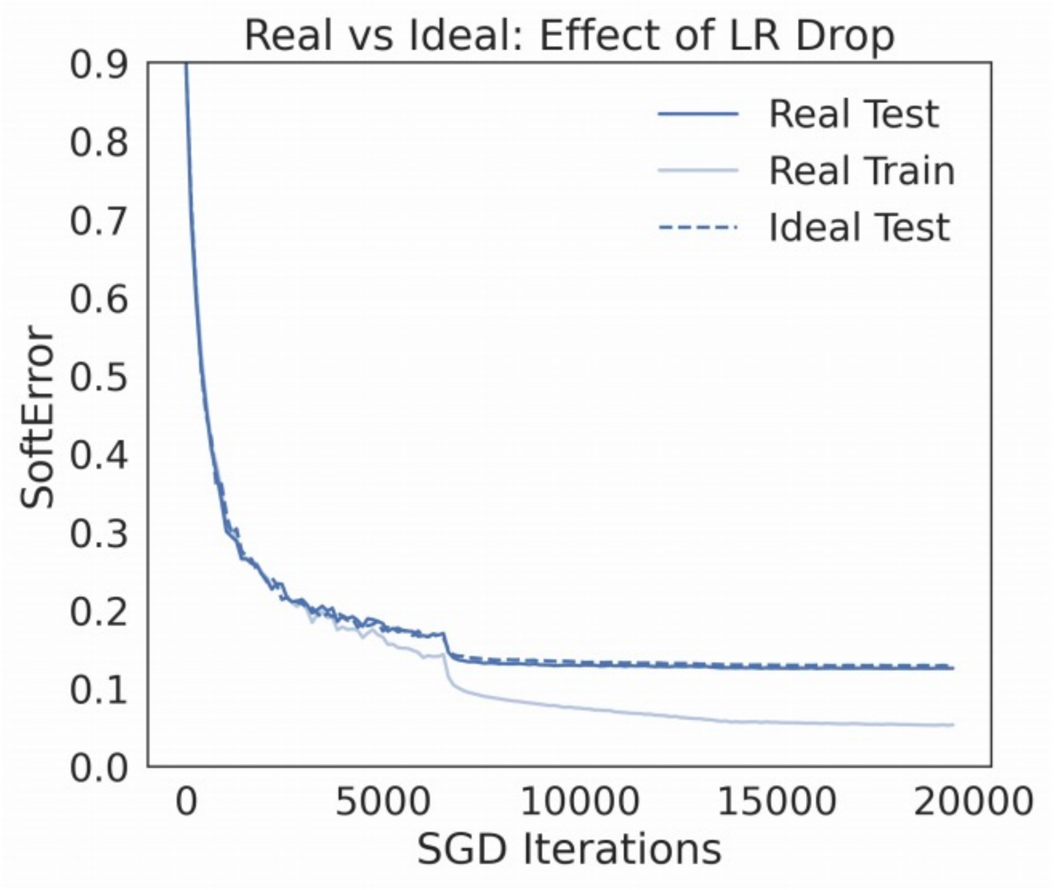
Both train to 0 Train Error, but convnet generalizes better.

Traditionally: due to “implicit bias” of SGD on the convnet.

Our view: due to better optimization in the Ideal World



Effect of Learning Rate



Concluding Thoughts

- Future models may be either **overparameterized** or **underparameterized** (GPT-3, T5, ResNeXt WSL)
    - Largest models **trained for less than one epoch**
    - Deep Bootstrap: understanding online optimization will be useful in either case

- Many arbitrary choices in deep learning (arch, loss, optimizer, activation..)
    - Which ones work for generalization?
    - Deep Bootstrap: Anything that works well for online optimization

- **Open Questions:**
    - Quantitative dependency of bootstrap-error on $(n, D, \mathcal{F}, t)$
    - Theoretical understanding? Toy models?

Extras

Choice of Metric

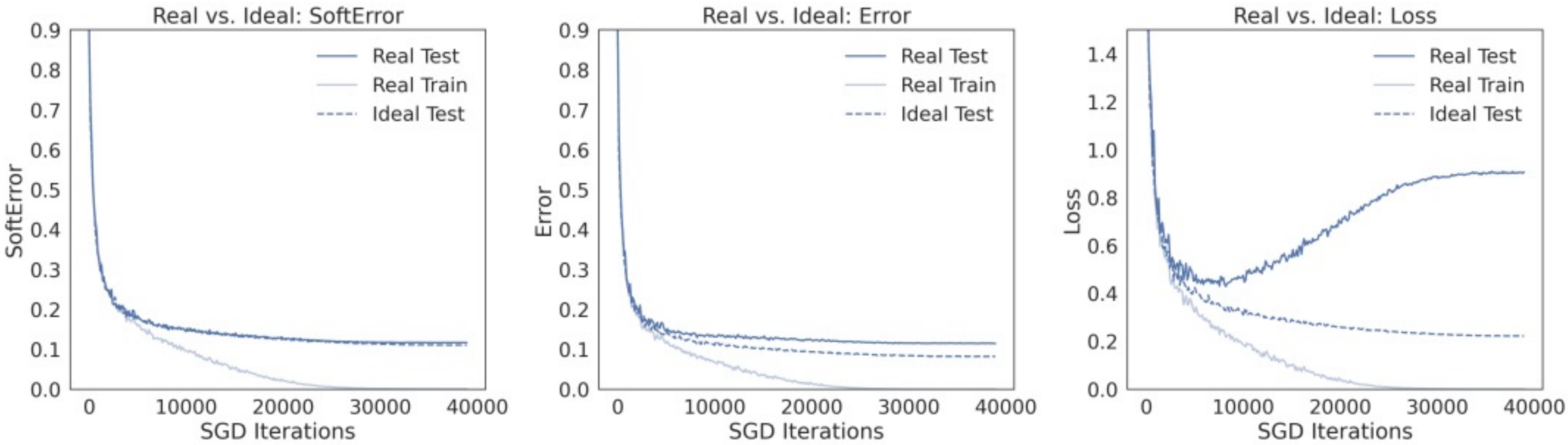


Figure 6: **SoftError vs. Error vs. Loss: ResNet-18.**

Why Soft-Error?

Want: RealWorld → IdealWorld as (model, data) → ∞.

- This doesn't always happen w.r.t Test Error.

Claim: In an overparameterized limit of (model, data) → ∞,

interpolating classifiers converge to *optimal samplers:* $f(x) \sim p(y|x)$

"Distributional Generalization" [Nakkiran, Bansal 2020]

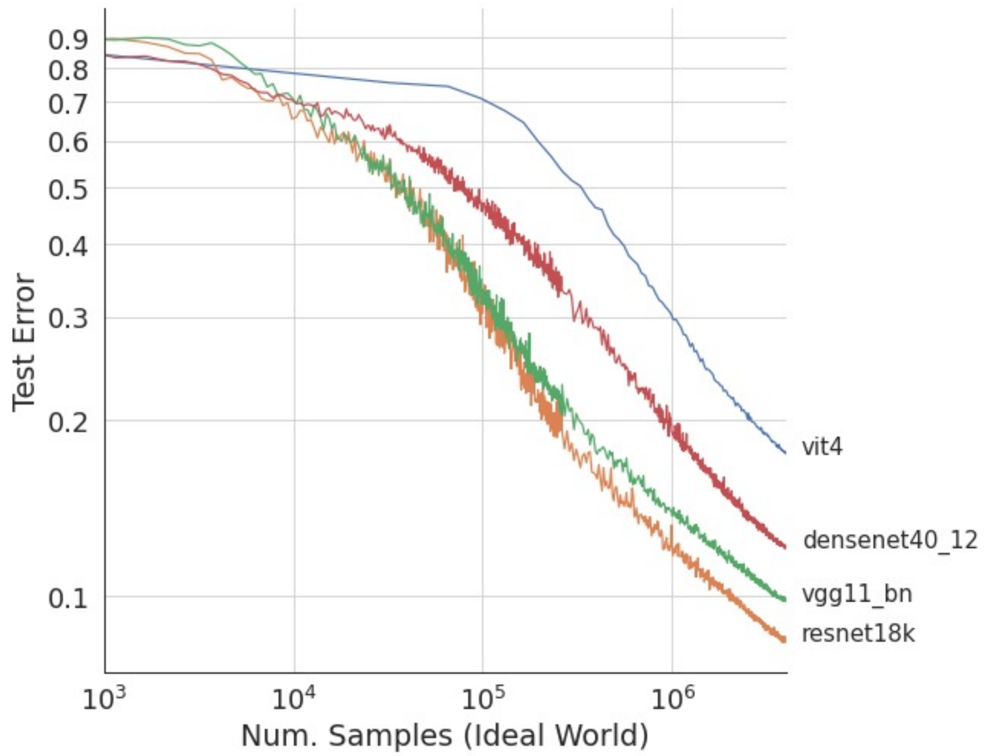
…**NOT** to Bayes-optimal classifiers: $f^*(x) = \operatorname{argmax}_y p(y|x)$

Scaling Laws in Ideal World

Study L(t) : Ideal-world learning curve

Empirically: power law

$L(t) \sim t^{-\alpha}$



What about Non-Deep Learning?

- Not true for well-specified linear regression!
- Can be contrived to be true for **misspecified** regression

$$x \sim \mathcal{N}(0, V)$$
$$y := \sigma(\langle \beta^*, x \rangle)$$

$$f_\beta(x) := \langle \beta, x \rangle$$

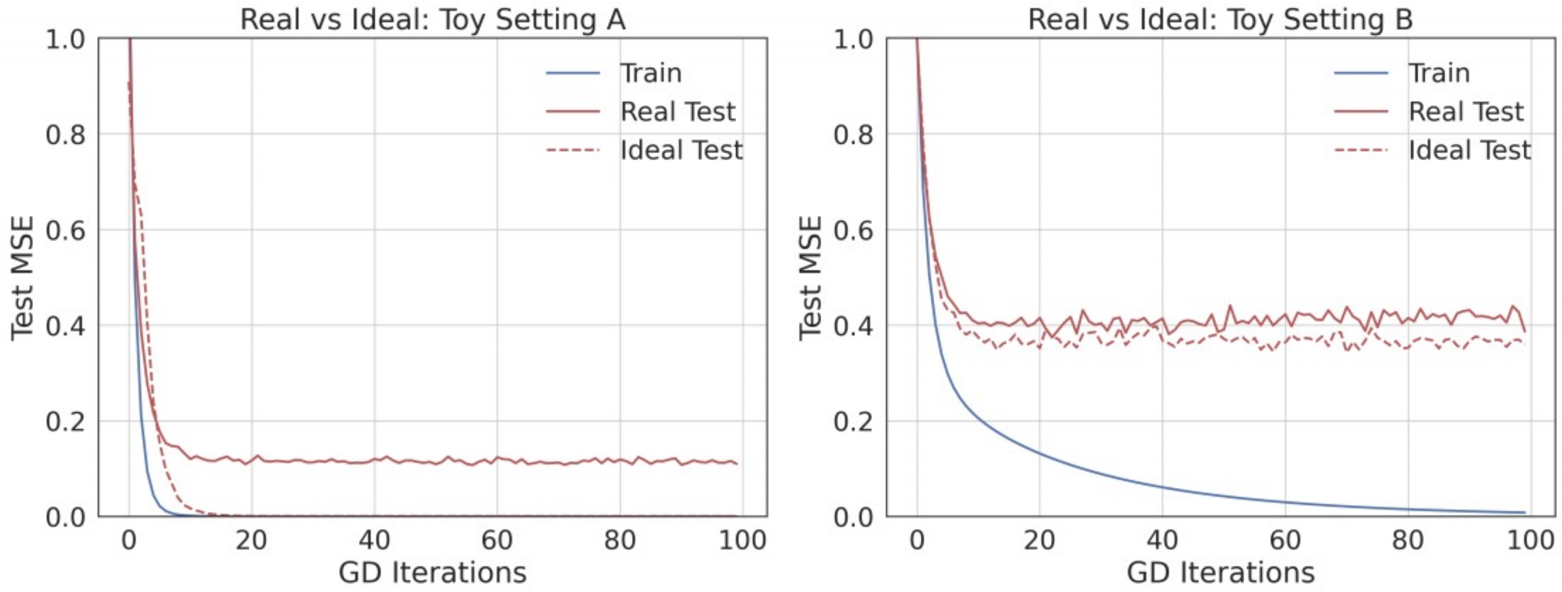
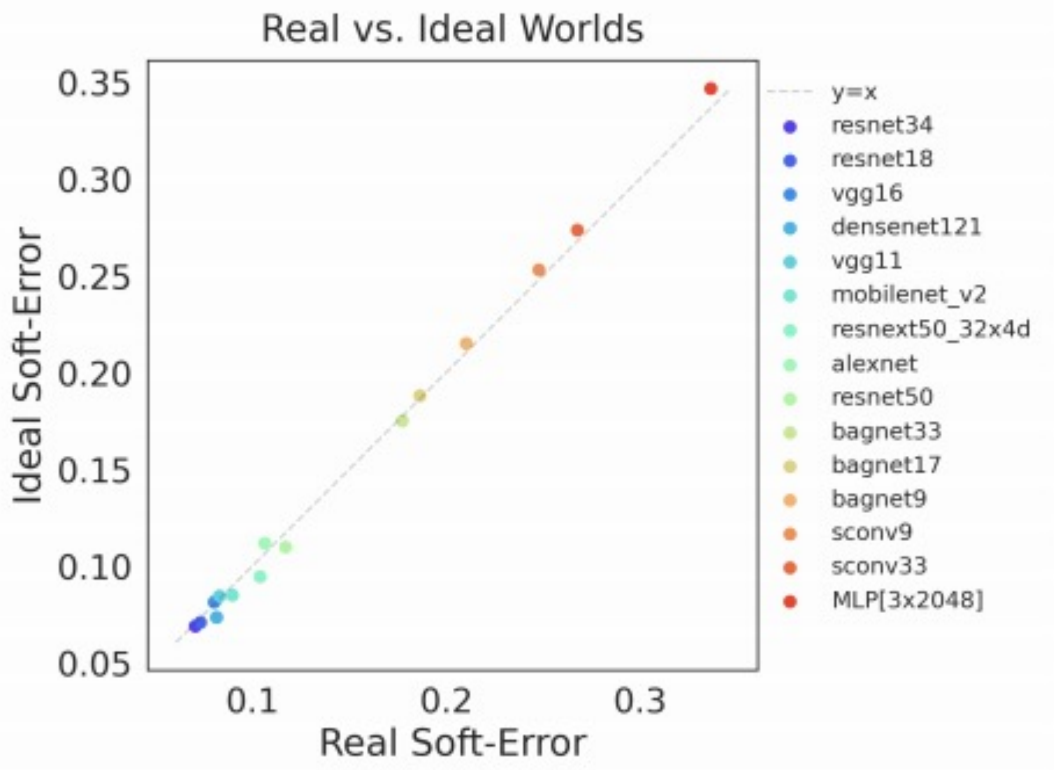


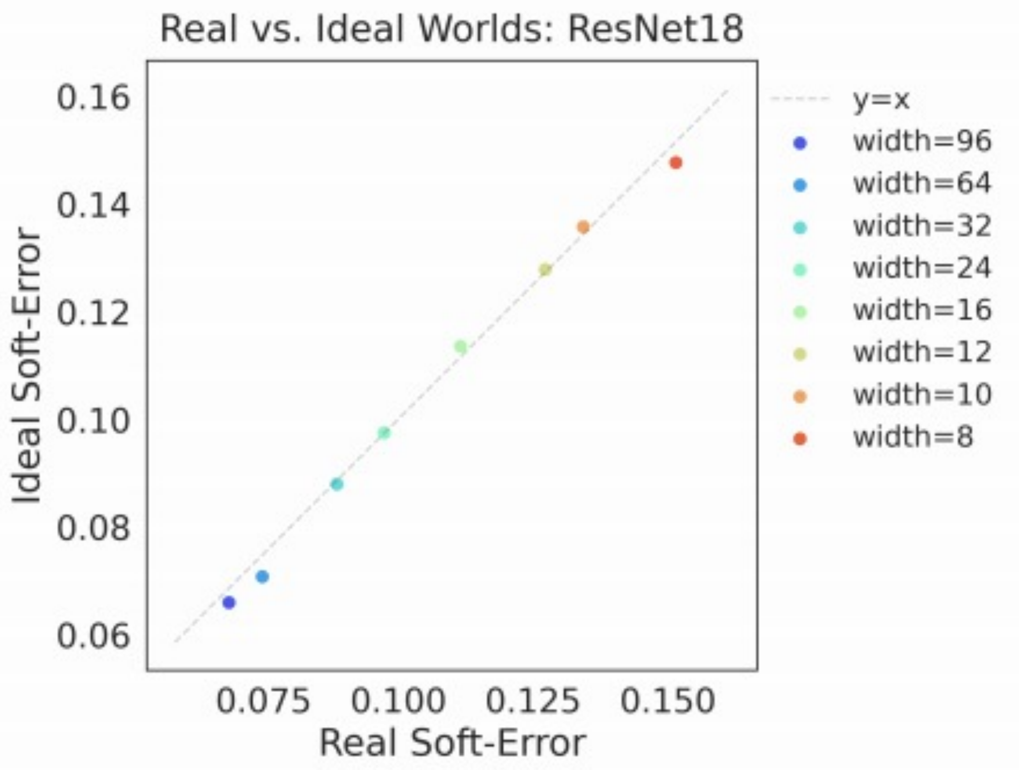
Figure 7: **Toy Example.** Examples of settings with large and small bootstrap error.

- **Setting A.** Linear activation $\sigma(x) = x$. With $n = 20$ train samples.
- **Setting B.** Sign activation $\sigma(x) = \text{sgn}(x)$. With $n = 100$ train samples.

ImageNet Experiments



(a) Standard architectures.

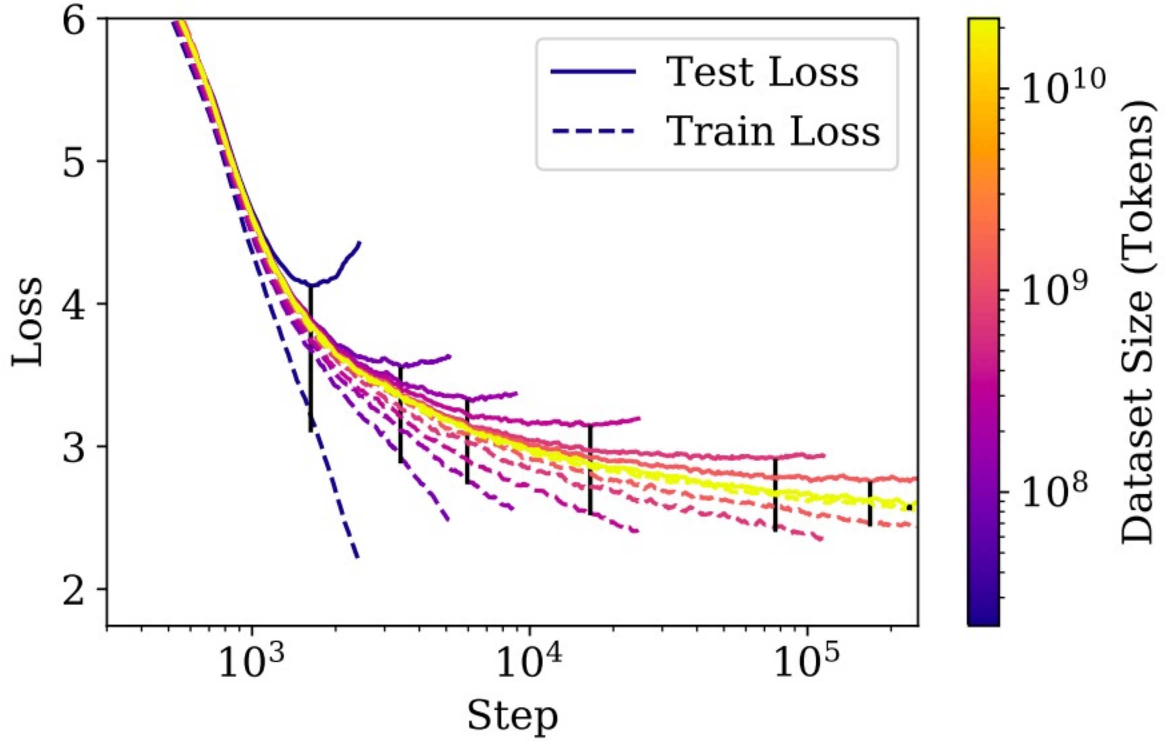


(b) ResNet-18s of varying width.

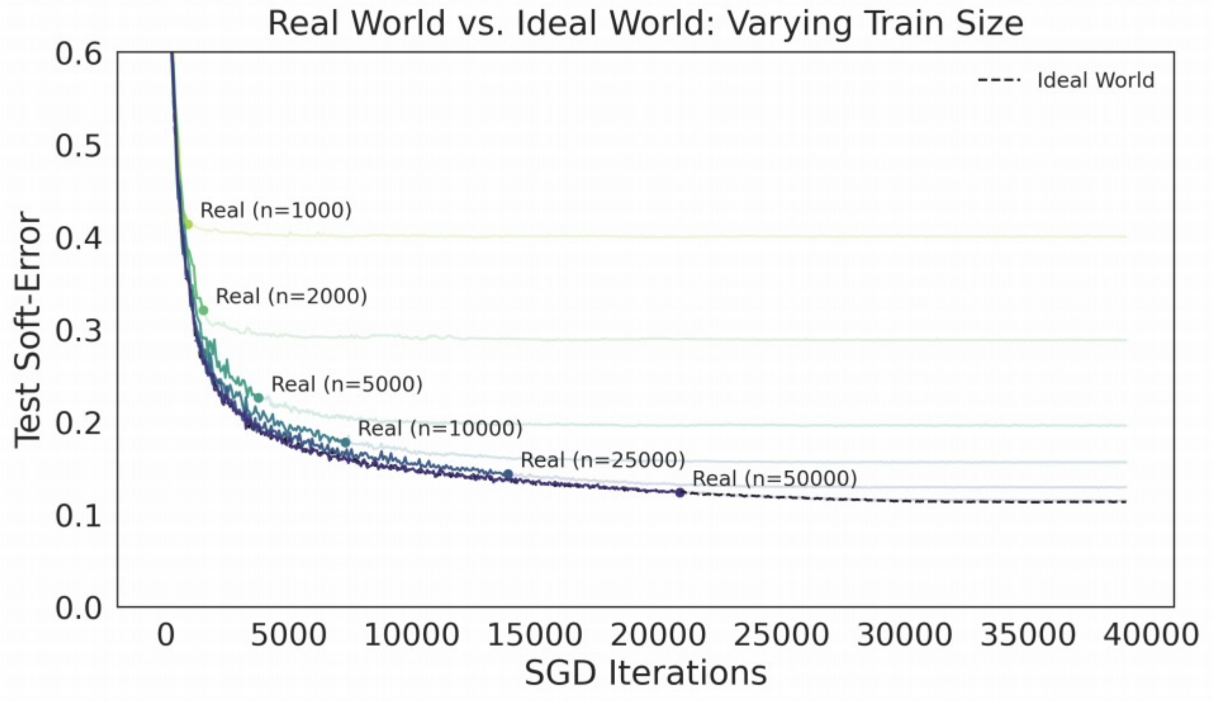
Figure 3: **ImageNet-DogBird.** Real World models trained on 10K samples.

GPT-3 Learning Curves

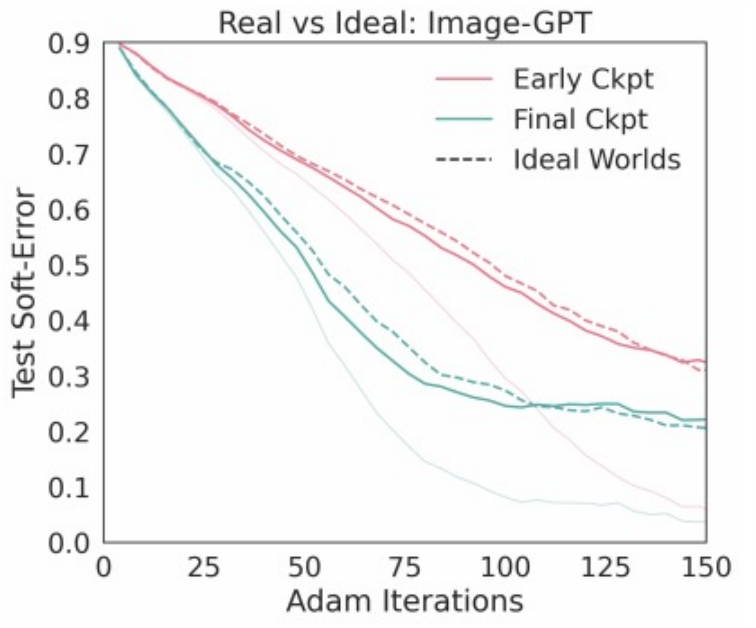
[Kaplan et al 2020]



ResNet18 Curves



Effect of Pretraining



(b) Pretrain: Image-GPT ($n = 2K$).

When Data-Aug Hurts

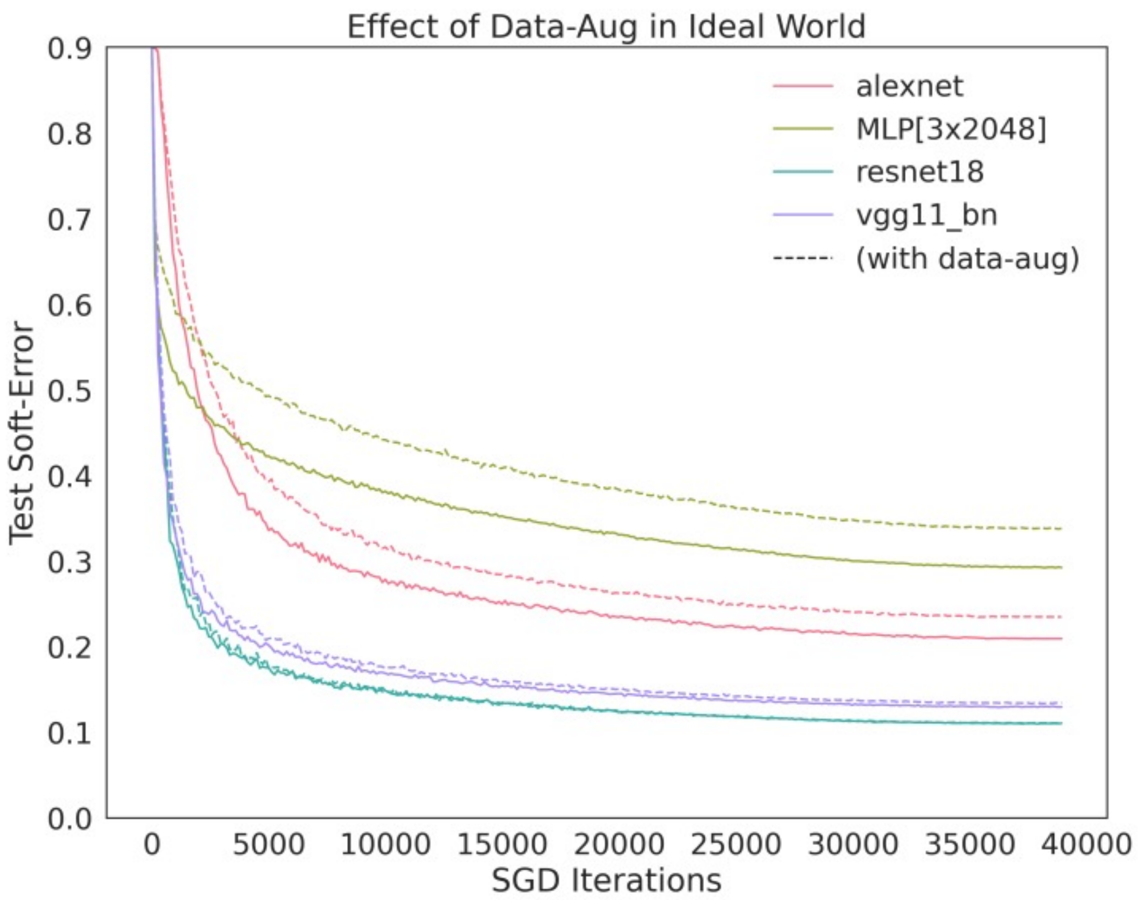


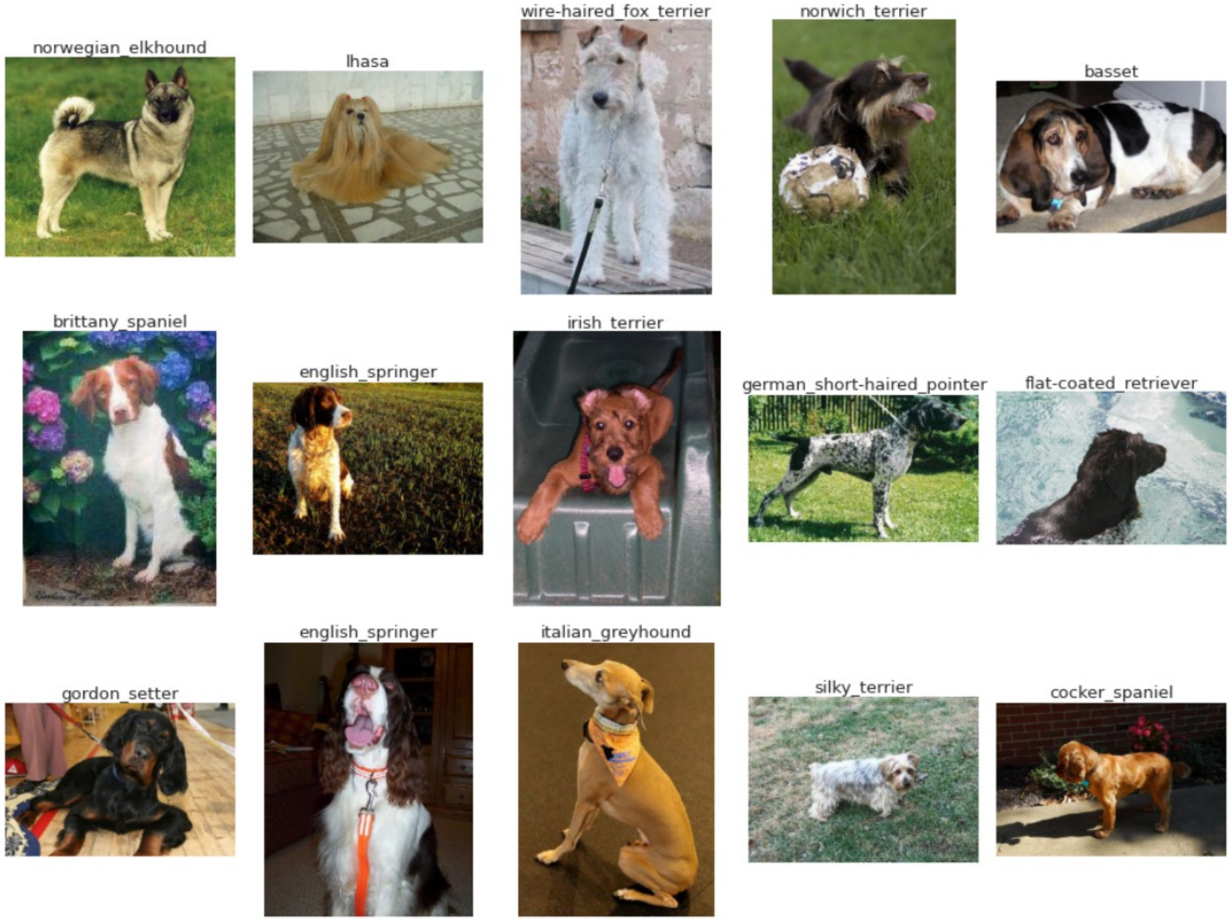
Figure 10: Effect of Data Augmentation in the Ideal World.

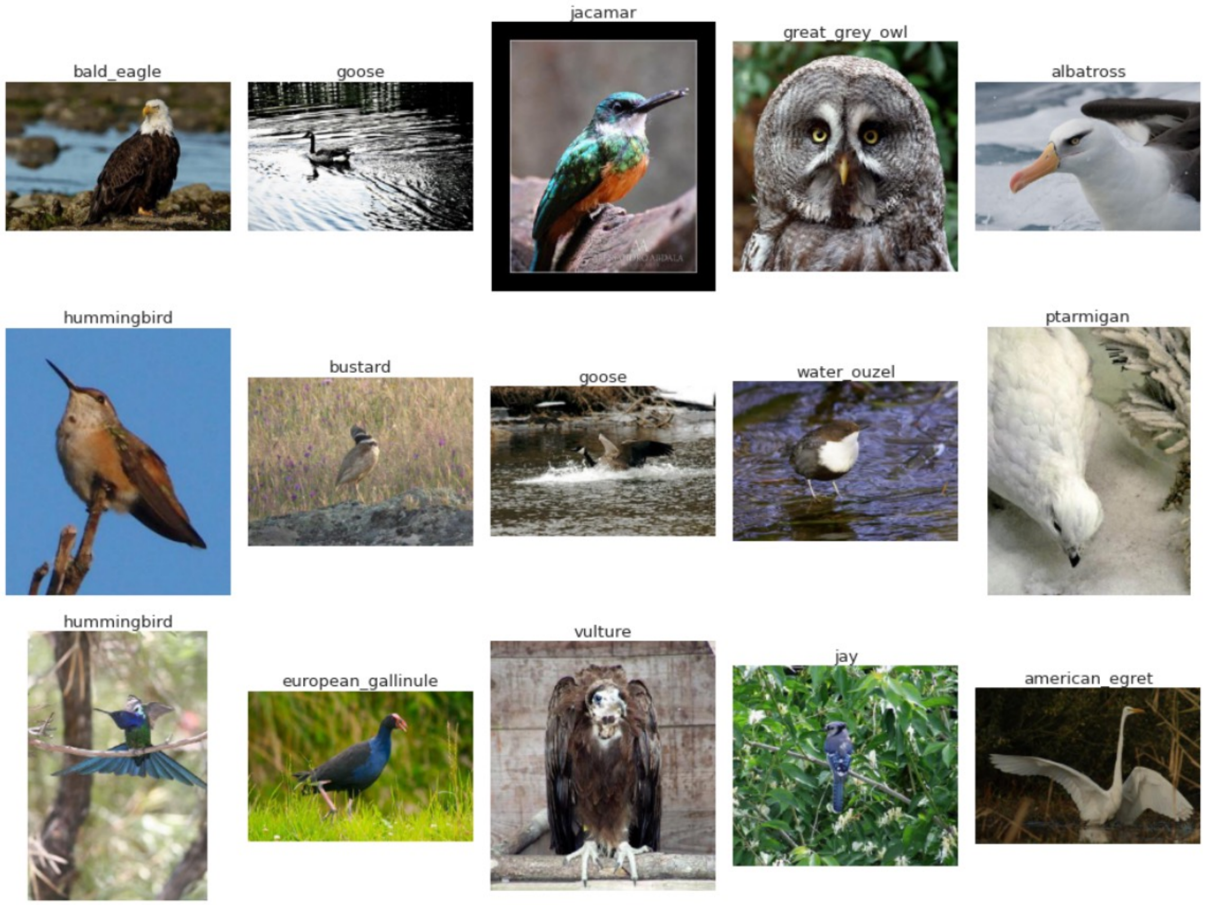
Figure 17: **CIFAR-5m Samples.** Random samples from each class (by row).

Figure 18: **CIFAR-10 Samples.** Random samples from each class (by row).

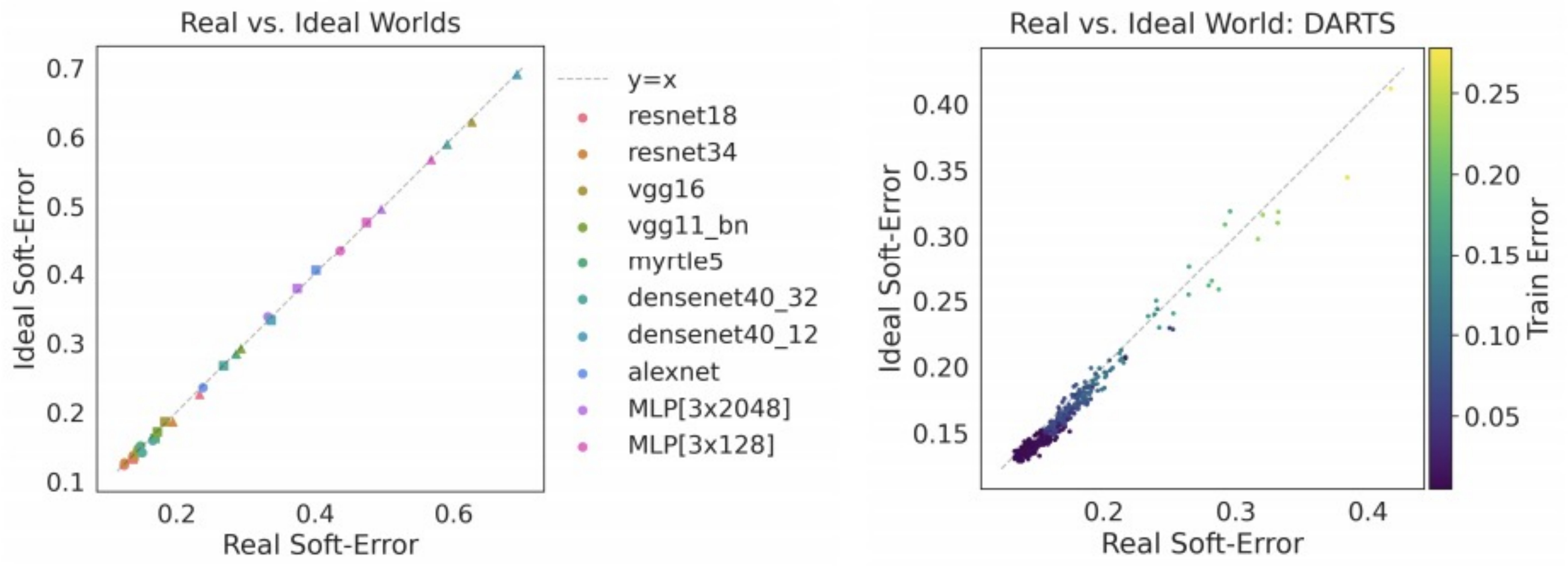
Trained On	Test Error On	
	CIFAR-10	CIFAR-5m
CIFAR-10	0.032	0.091
CIFAR-5m	$\boxed{0.088}$	0.097

Table 2: WRN28-10 + cutout on CIFAR-10/5m


norwegian_elkhound
lhasa
wire-haired_fox_terrier
norwich_terrier
basset
brittany_spaniel
english_springer
irish_terrier
german_short-haired_pointer
flat-coated_retriever
gordon_setter
english_springer
italian_greyhound
silky_terrier
cocker_spaniel


bald_eagle
goose
jacamar
great_grey_owl
albatross
hummingbird
bustard
goose
water_ouzel
ptarmigan
hummingbird
european_gallinule
vulture
jay
american_egret

CIFAR-5m Experiments

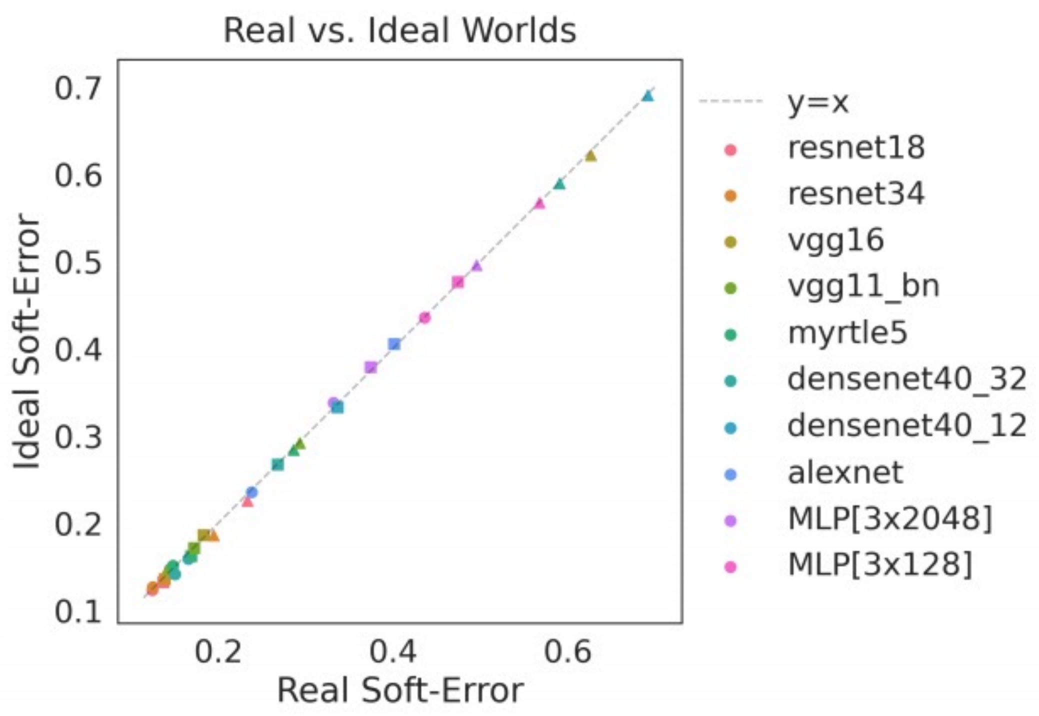


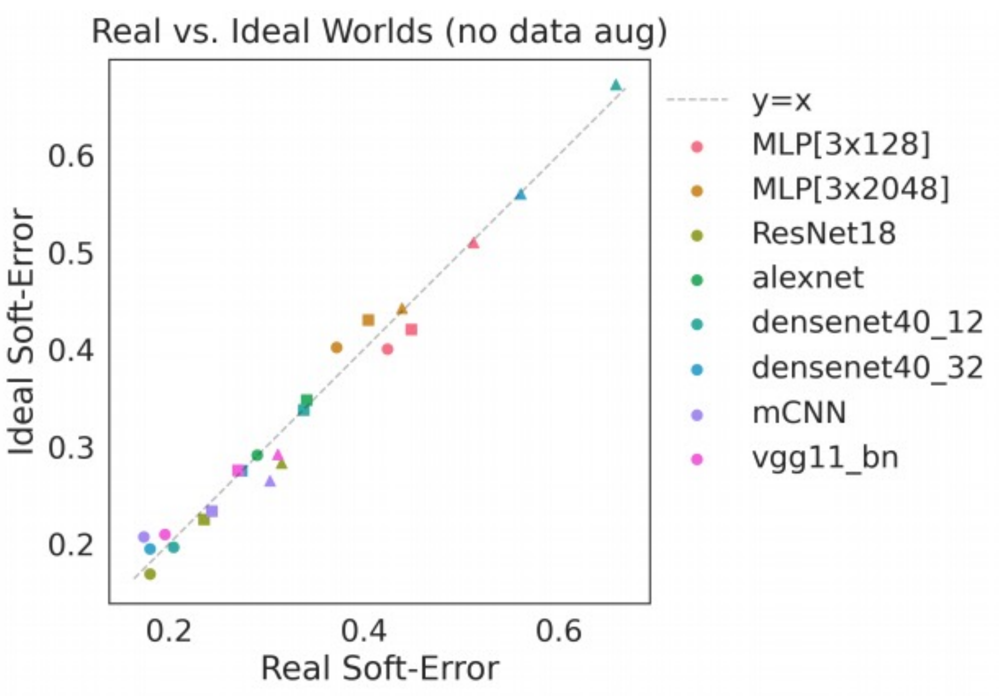
(a) Standard architectures.

(b) Random DARTS architectures.

Figure 2: **Real vs Ideal World: CIFAR-5m.** SGD with 50K samples. (a): Varying learning-rates 0.1 (●), 0.01 (■), 0.001 (▲). (b): Random architectures from DARTS space (Liu et al., 2019).

ImageNet Experiments





Validation: Summary of Experiments

- **CIFAR-5m:** 5-million synthetic samples from a generative model trained on CIFAR-10
  - Realistic: Training WRN on n=50K from CIFAR-5m yields 91.2% test acc on CIFAR-10
- **ImageNet-DogBird:** 155K images by collapsing ImageNet catagories.
  - Real World: n=10K for 120 epochs
  - Ideal World: n=155K for < 8 epochs (approximation of $n = \infty$)
- **Various archs:** convnets, ResNets, MLPs, Image-GPT, Vision-Transformer

Classical Framework (ERM)

Classical Framework: Finite data, need to understand *generalization gap*

$$\underline{\mathrm{TestError}(f_t)} = \underline{\cancelto{0}{\mathrm{TrainError}(f_t)}} + \underbrace{[\mathrm{TestError}(f_t) - \cancelto{0}{\mathrm{TrainError}(f_t)}]}_{\text{Generalization gap}}$$

"Good models are those with small generalization gap"

Obstacles:

1. Hard: Decades of work, little progress.
2. Large models can fit train sets → trivializes framework