Predicting Deep Neural Network Generalization with Perturbation Response Curves

Yair Schiff, Brian Quanz,
Payel Das, and Pin-Yu Chen

NeurIPS 2021 Poster Session

NEURAL INFORMATION PROCESSING SYSTEMS

IBM Research

Problem statement and background: Predicting neural network generalization

- There is a gap in the literature for an **efficient and intuitive measure that can predict generalization of a deep neural network**

- Predicting Generalization in Deep Learning (PGDL) NeurIPS 2020 encouraged participants to provide ***complexity*** **measures calculated from network weights and training data to predict generalization gaps**

Y. Jiang, P. Foret, S. Yak, D. M. Roy, H. Mobahi, G. K. Dziugaite, S. Bengio, S. Gunasekar,I. Guyon, and B. Neyshabur, “Neurips 2020 competition: Predicting generalization in deeplearning,”arXiv preprint arXiv:2012.07976, 2020

Core idea

User inputs

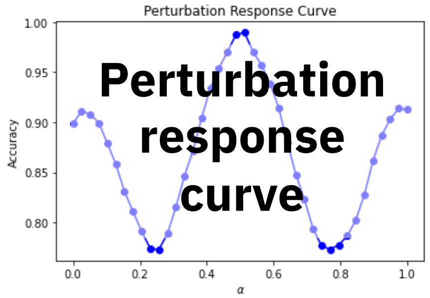
1. Trained model

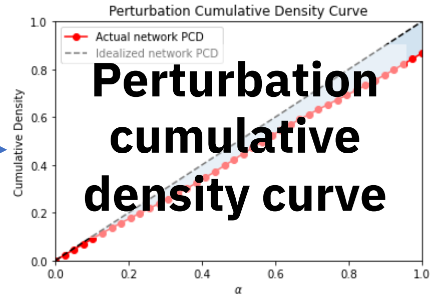
2. Training data

3. Parametric transformation

$\mathcal{T}_\alpha$

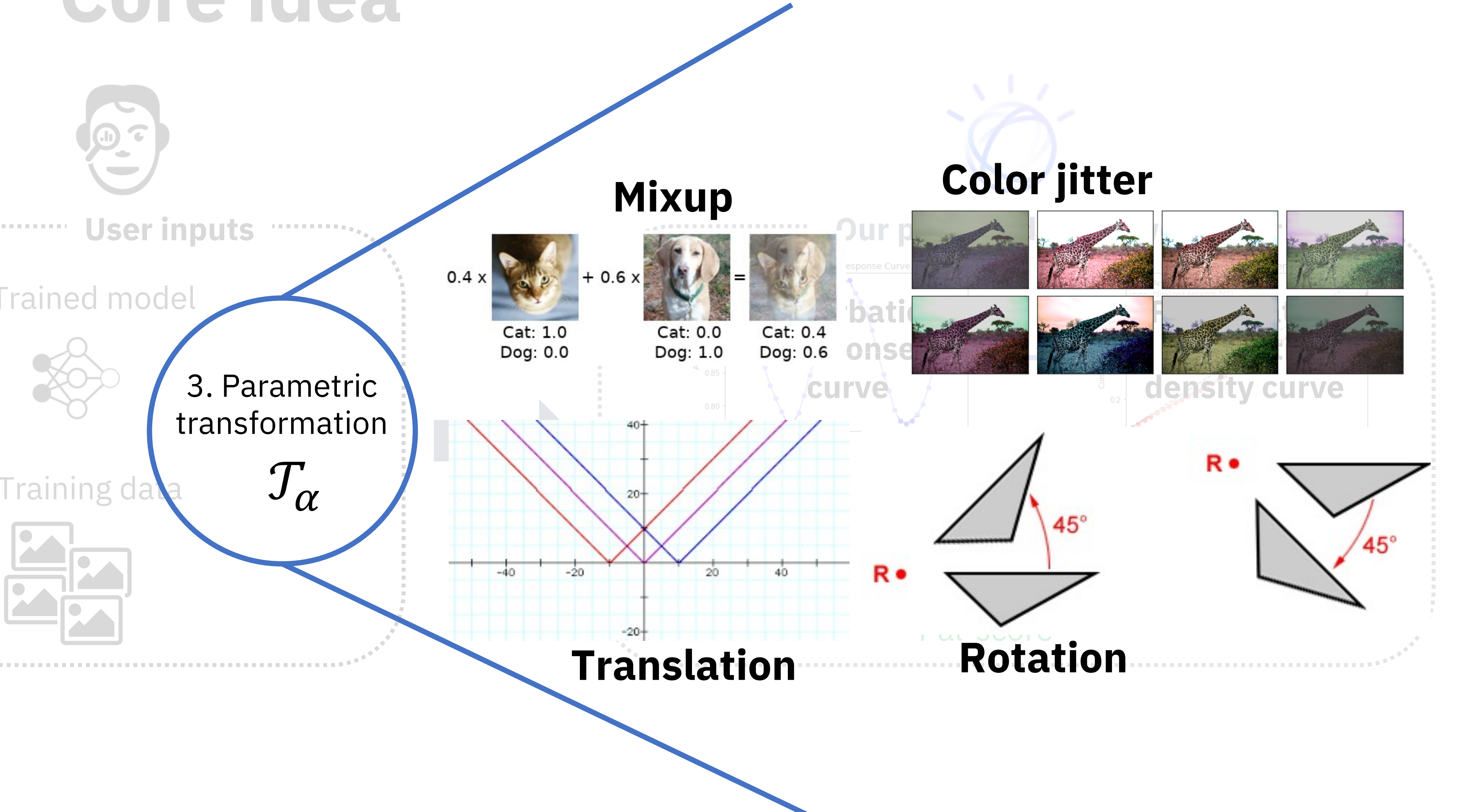
Our proposed framework





Output:
Gi-score
&
Pal-score

Core idea


User inputs
1. Trained model
2. Training data
3. Parametric transformation
$\mathcal{T}_\alpha$
Mixup
0.4 x
+ 0.6 x
=
Cat: 1.0
Dog: 0.0
Cat: 0.0
Dog: 1.0
Cat: 0.4
Dog: 0.6
Color jitter
Translation
Rotation
R
45°
R
45°

Core idea

User inputs

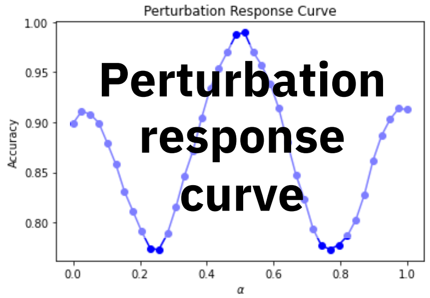
1. Trained model

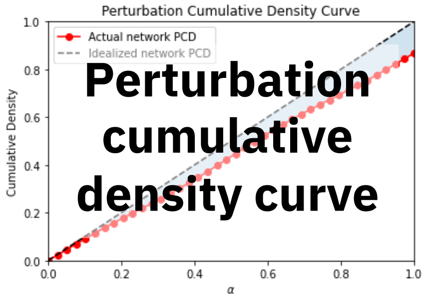
2. Training data

3. Parametric transformation

$\mathcal{T}_\alpha$

Our proposed framework



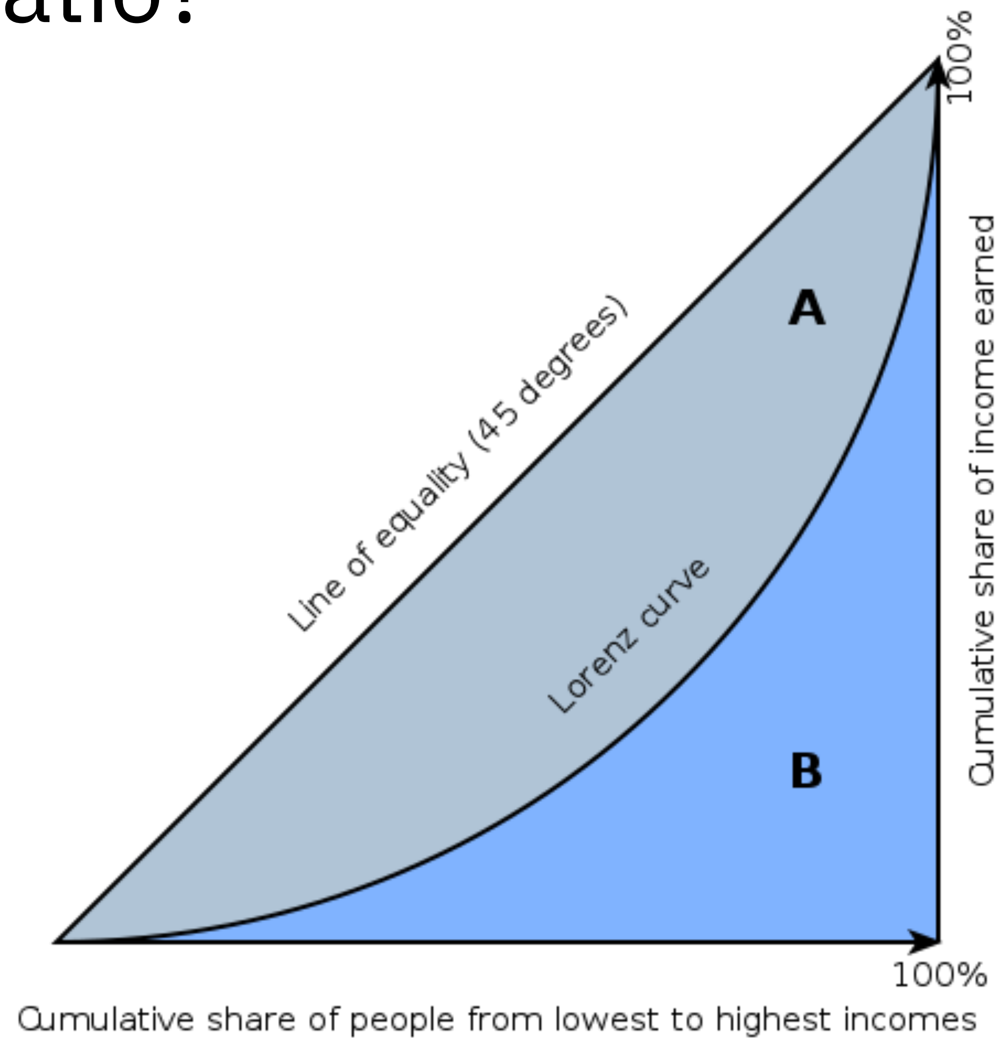


Output:
Gi-score
&
Pal-score

Use cases and value

- This framework will be useful for **data scientists** and **machine learning practitioners**
- Useful for **predicting generalization and robustness** and defining new **regularization** approaches
- Our work can therefore serve as a **model selection criterion,** similar to R^2 and other related statistics

Interlude: What are the Gini coefficient and Palma ratio?



Source: https://en.wikipedia.org/wiki/File:Economics_Gini_coefficient2.svg

Detailed description: Step 1 – Calculate Perturbation Response curve

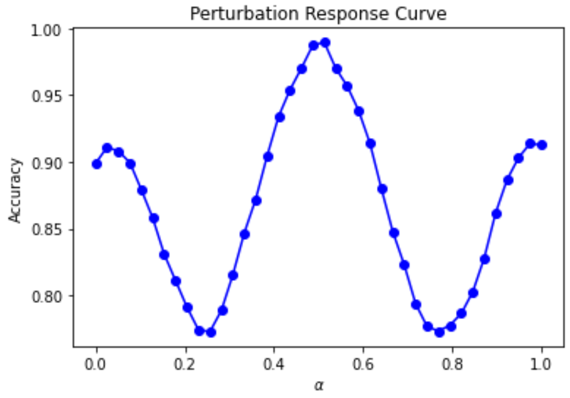
---

Algorithm 1: Building Perturbation Response (PR) Curve

Inputs: Trained model f; Dataset $\mathcal{D}$; Perturbation $\mathcal{T}_\alpha$; Min perturbation magnitude $\alpha_{\min}$; Max perturbation magnitude $\alpha_{\max}$; Number of perturbation magnitudes to measure n_p; Layer at which to apply the perturbation ℓ; number of batches to sample n_b; batch size b_s

Output: PR Curve: Arrays of regularly spaced perturbation magnitudes ranging from $\alpha_{\min}$ to $\alpha_{\max}$ of length n_p $[\alpha_{\min}, \alpha_{\max}][n_p]$ and accuracy array at each perturbation magnitude of length n_p $\mathcal{A}_\alpha[n_p]$

for $i \leftarrow 0$ *to* $n_p - 1$ **do**
- $\alpha_i \leftarrow [\alpha_{\min}, \alpha_{\max}][i]$
- Shuffle $\mathcal{D}$
- **for** $k \leftarrow 0$ *to* $n_b - 1$ **do**
  - $\mathcal{D}_{sample} \leftarrow \mathcal{D}[kb_s : (k+1)b_s]$ `// batch k of D`
  - $\mathcal{A}^{(\ell)}_{\alpha_i}[\mathrm{k}] \leftarrow$ batch accuracy under perturbation $\mathcal{T}_{\alpha_i}$ (Equation 1)
- $\mathcal{A}_\alpha[\mathrm{i}] \leftarrow \sum_k \mathcal{A}^{(\ell)}_{\alpha_i}[\mathrm{k}]/n_b$



- Works with any trained model, image dataset, and parametric perturbation
- Can be applied at any depth of a neural network

Detailed description: Step 2 – Calculate Gi and Pal scores

Algorithm 2: Gi-Score computation given PR Curve for a model

Inputs: Arrays of perturbation magnitude $\alpha[n]$ and accuracy $\mathcal{A}_\alpha[n]$

Output: Gi-score gi

$a_t[0] \leftarrow 0$ `// initialize 1st element of trapezoidal areas array with 0`

for $i \leftarrow 0$ *to* $n-2$ **do**

  $a_t[i+1] \leftarrow 0.5(\alpha[i+1]-\alpha[i])(\mathcal{A}_\alpha[i]+\mathcal{A}_\alpha[i+1])$

for $i \leftarrow 1$ *to* $n-1$ **do**

  $a_t[i] \leftarrow a_t[i]+a_t[i-1].$ `// cumulative sum`

$d[i]=\alpha[i]-a_t[i], \forall i$

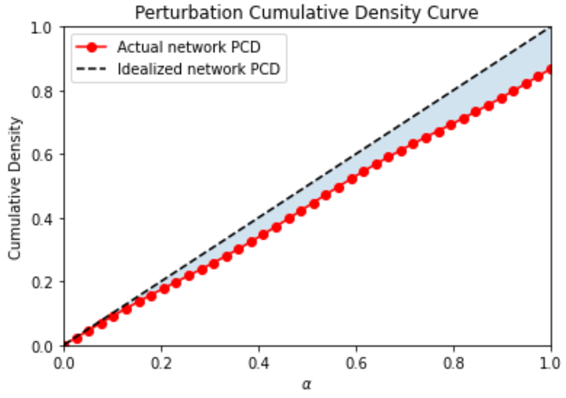
$gi = 0$

for $i \leftarrow 0$ *to* $n-2$ **do**

  $gi \leftarrow gi + 0.5(\alpha[i+1]-\alpha[i])(d[i]+d[i+1])$

$gi \leftarrow gi/(0.5\alpha[n-1]^2)$ `// Divide by area under line of equality`

return gi



PGDL results: We outperform winning team from PGDL competition in majority of tasks & overall

	CIFAR-10		SVHN	CINIC-10		Oxford Flowers	Oxford Pets	Fashion MNIST	*All Avg*
	VGG	*NiN*	*NiN*	*Conv w/bn*	*Conv w/o bn*	*NiN*	*NiN*	*VGG*	
Single measures only									
Gi *inter* ℓ=0	3.03	**34.34**	26.58	21.01	6.96	33.05	**18.46***	4.48	18.49
Gi *inter* ℓ=1	**7.88**	22.59	12.17	12.58	8.39	7.52	4.68	**16.16***	11.49
Pal *inter* ℓ=0	3.14	26.39	24.25	21.11	6.37	29.62	15.96	4.21	16.38
Pal *inter* ℓ=1	7.31	12.75	9.79	12.09	7.71	6.37	3.46	14.13	9.20
Gi *intra* ℓ=0	0.84	30.54	**41.75***	22.97	11.46	**42.44**	16.21	5.10	**21.41**
Gi *intra* ℓ=1	0.22	17.18	10.96	9.50	12.43	6.92	3.60	5.55	8.29
Pal *intra* ℓ=0	0.61	24.36	31.82	24.15	11.01	38.10	14.04	5.12	18.65
Pal *intra* ℓ=1	0.44	10.34	13.48	8.68	11.09	5.88	3.02	6.25	7.40
Mixup	0.03	14.18	22.75	**30.30**	**19.51**	35.30	9.99	7.75	17.48
Mani. Mixup	2.24	2.88	12.11	4.23	4.84	0.03	0.13	0.19	3.33
Combination measures									
PCA Gi&Mix.	0.04	33.16	38.08	**33.76***	**20.33***	40.06	13.19	**10.30**	**23.62***
Pal ℓ=0*ℓ=1	1.71	**35.77***	**41.58**	25.14	9.50	38.92	**18.41**	5.61	22.08
Pal *inter+intra*	**24.84***	29.70	14.04	1.64	3.45	14.84	2.13	4.89	11.94
DBI*Mixup[1]	0.00	25.86	32.05	31.79	15.92	**43.99***	12.59	9.24	21.43

P. Natekar and M. Sharma, “Representation based complexity measures for predicting generalization in deep learning,” 2020

Example PR Curve Pairs: Mixup $\boldsymbol{\alpha}$=0.5 not enough to differentiate them

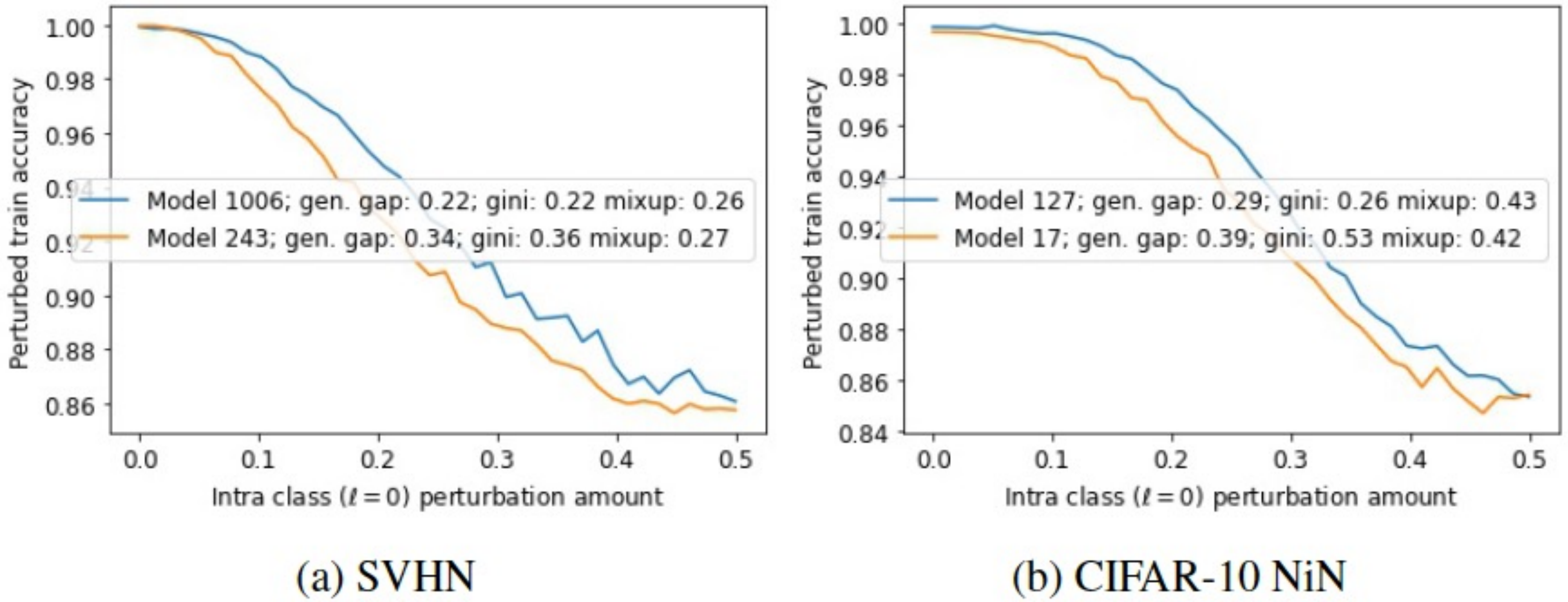
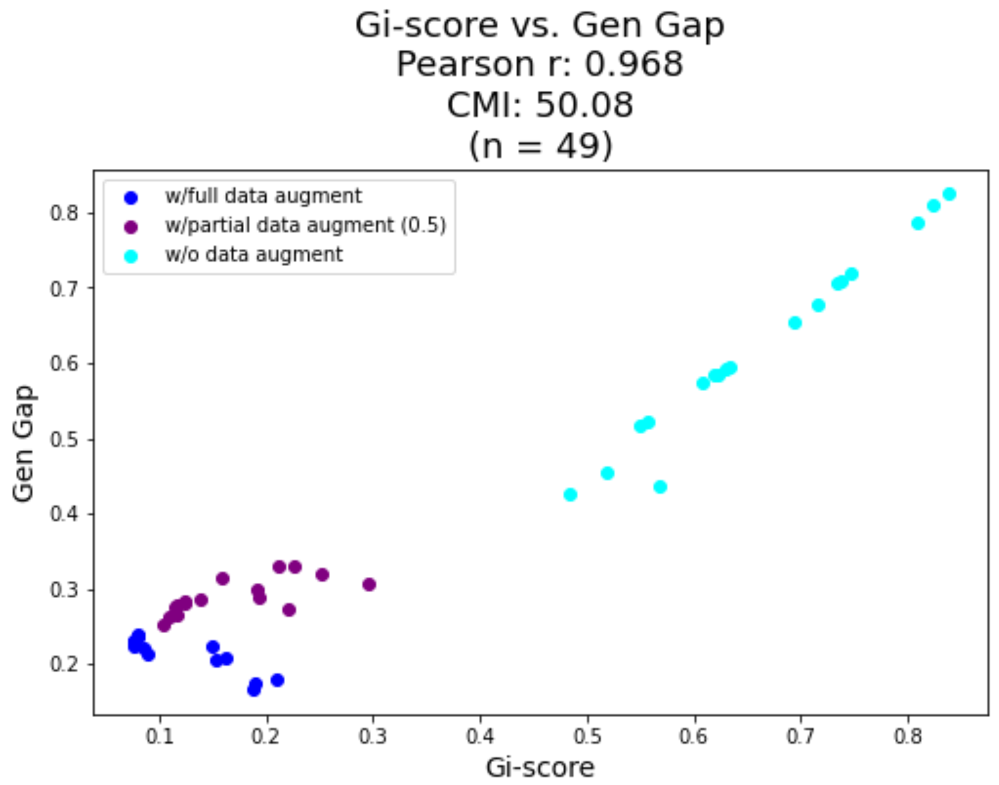


Figure 2: Examples of PR Curves with normalized scores and gen. gaps for 2 different models showing different performance fall-off captured by Gi intra score, but mixup scores roughly the same.

Measuring invariance: experimental setup

	CIFAR-10	SVHN
Rotation	(-180, 179)	(-90, 90)
Horizontal translation	(-0.5, 0.5)	(-0.5, 0.5)
Vertical translation	(-0.5, 0.5)	(-0.5, 0.5)
Color jittering	(-0.25, 0.25)	(-0.25, 0.25)

Table 2: Perturbation minimum and maximum magnitudes by perturbation type and dataset. Minimum and maximums are displayed in each cell as an ordered pair.



Measuring invariance: We accurately predict degree of invariance across perturbation types

	CIFAR-10 Resnet	CIFAR-10 VGG	SVHN Resnet	SVHN VGG
Rotation	$(n = 34)$	$(n = 93)$	$(n = 49)$	$(n = 142)$
Acc. on augmented train subset	27.99	**16.99**	47.24	42.97
Mean acc. on PR curve	27.61	15.61	48.14	44.05
Gi-score	**41.54**	15.29	**54.11**	**46.11**
Horizontal translation	$(n = 36)$	$(n = 112)$	$(n = 50)$	$(n = 143)$
Acc. on augmented train subset	41.79	33.48	29.49	24.20
Mean acc. on PR curve	45.03	33.00	29.88	24.28
Gi-score	**50.07**	**34.31**	**34.56**	**25.94**
Vertical translation	$(n = 36)$	$(n = 107)$	$(n = 49)$	$(n = 141)$
Acc. on augmented train subset	26.79	35.68	51.88	50.98
Mean acc. on PR curve	26.55	37.33	52.39	52.22
Gi-score	**34.85**	**39.07**	**59.02**	**52.83**
Color-jittering	$(n = 44)$	$(n = 130)$	$(n = 49)$	$(n = 143)$
Acc. on augmented train subset	35.77	28.44	43.08	**30.07**
Mean acc. on PR curve	39.12	29.37	43.26	28.67
Gi-score	**44.63**	**30.79**	**50.08**	29.45

Summary

- We propose a **flexible framework** that provides high quality prediction of a trained neural network's generalization capability
- We provide multiple **new and efficient neural network generalization predictors**: Gi-score, Pal-score, and their combinations
- Our work **can be used with any parametric transformation** to compare the degree to which a network is invariant to that transformation

Thank you!