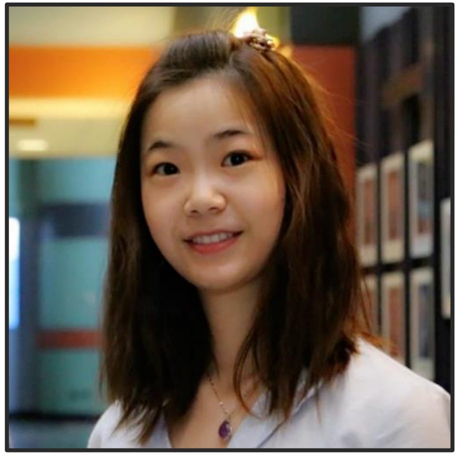
NVIDIA

Angular Visual Hardness

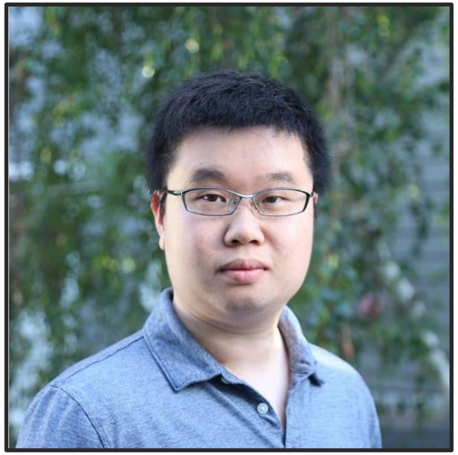
Zhiding Yu   Machine Learning Group, NVIDIA Research

zhidingy@nvidia.com



Beidi Chen, Rice

Weiyang Liu, Georgia Tech



Zhiding Yu, NVIDIA

Jan Kautz, NVIDIA

Anshumali Shrivastava, Rice

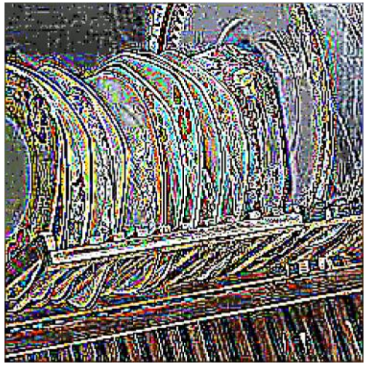
Animesh Garg, NVIDIA

Anima Anandkumar, NVIDIA

Human Visual Hardness

plate rack

sharpness



contrast

blur

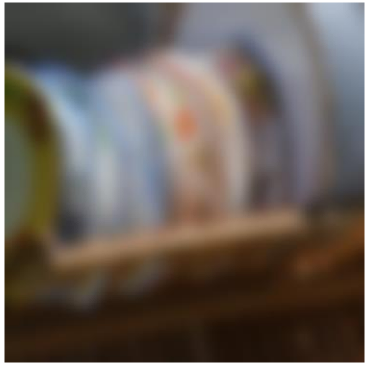


Image Degradation

dishwasher

saltshaker

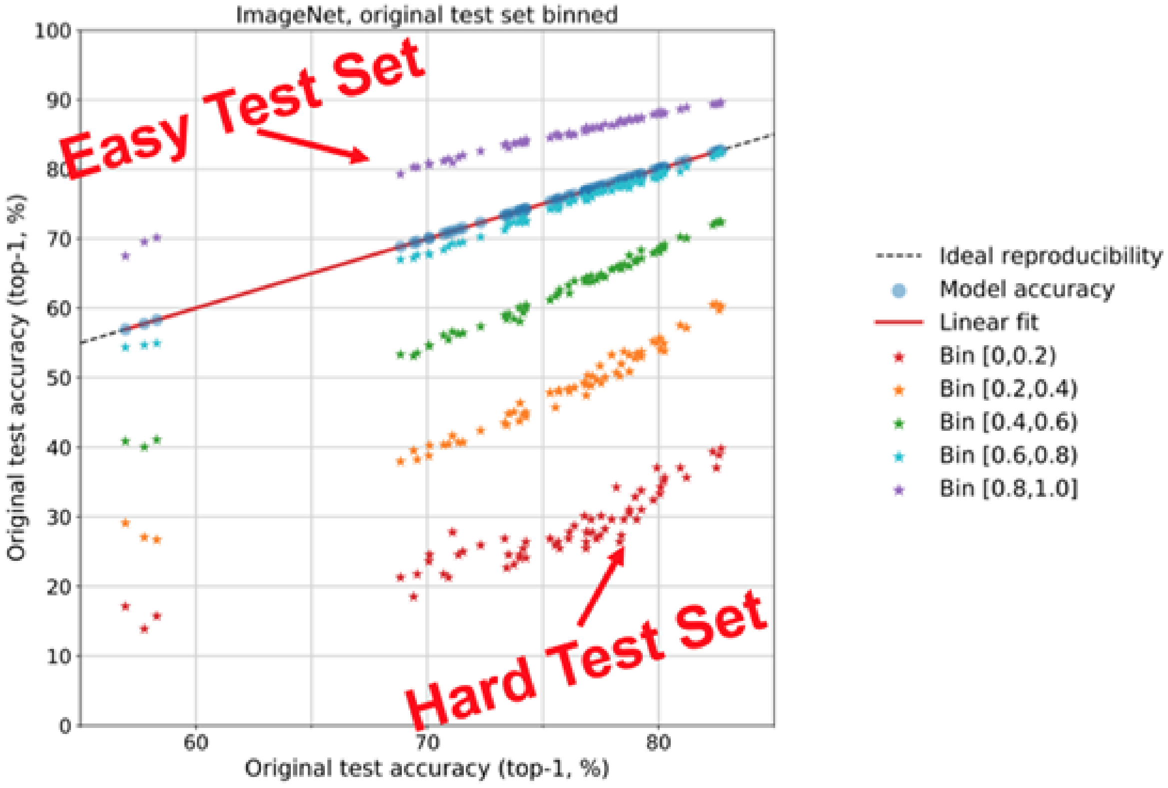
nail

oil filter

Semantic Ambiguity

Human Selection Freq (HSF): A Visual Hardness Proxy

Human Labeling Interface



Recht et al. “Do ImageNet Classifiers Generalize to ImageNet?” ICML 2019

Gap between Human Recognition and CNNs

Hard for Human but Easy for CNNs

Easy for Human but Hard for CNNs

	Nail	Oil Filter	Golf Ball	Radio
Softmax	0.93	0.998	0.001	0.001
HSF	0.2	0.2	1.0	1.0

Softmax Cross-Entropy Loss

$$L = \frac{1}{N}\sum_i L_i = \frac{1}{N}\sum_i -\log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right)$$

Magnitude Information

Angle Information

$$L_i = -\log\left(\frac{e^{\|\boldsymbol{W}_{y_i}\|\|\boldsymbol{x}_i\|\cos(\theta_{y_i})}}{\sum_j e^{\|\boldsymbol{W}_j\|\|\boldsymbol{x}_i\|\cos(\theta_j)}}\right)$$

Model Confidence

Angular Visual Hardness (AVH)

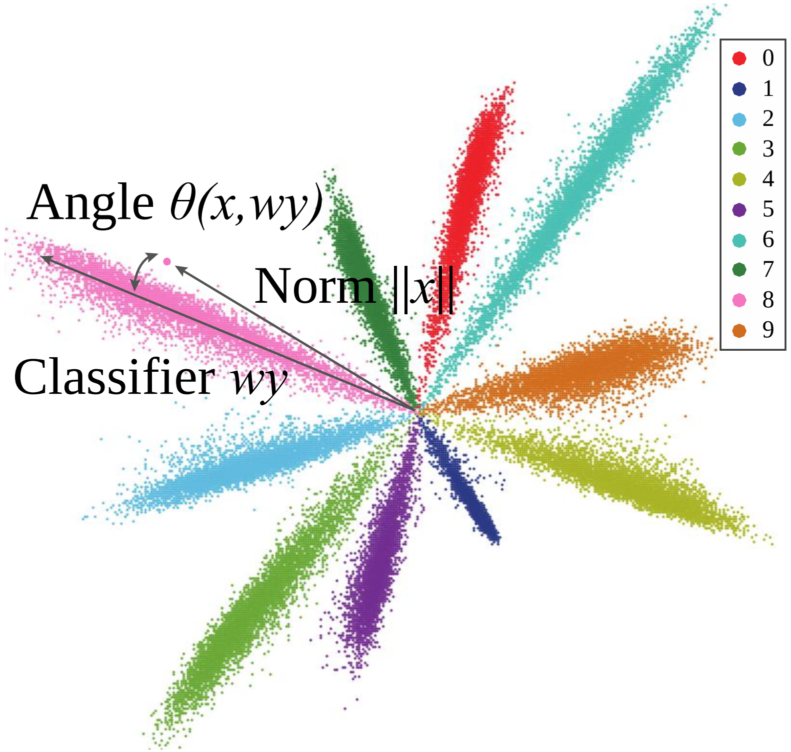
Given a sample *x* with label *y*:

$$\boldsymbol{AVH}(x) = \frac{\mathcal{A}(x, w_y)}{\sum_{i=1}^{C} \mathcal{A}(x, w_i)}$$

where,

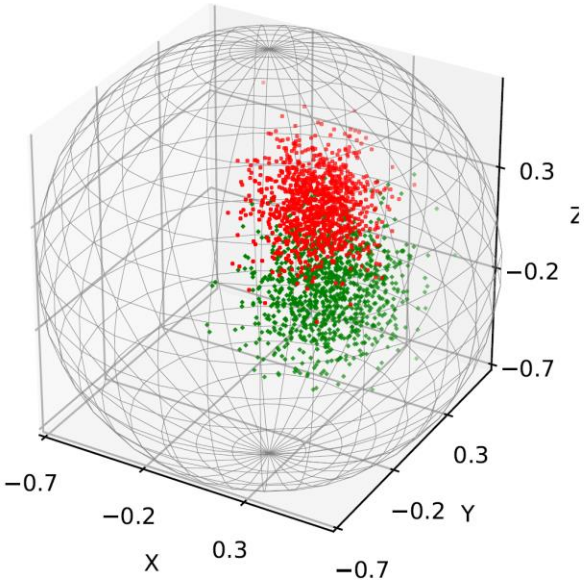
$$\mathcal{A}(\boldsymbol{u}, \boldsymbol{v}) = \arccos\left(\frac{\langle \boldsymbol{u}, \boldsymbol{v} \rangle}{\|\boldsymbol{u}\| \|\boldsymbol{v}\|}\right)$$

w_i is the classifier for the *i*-th class.

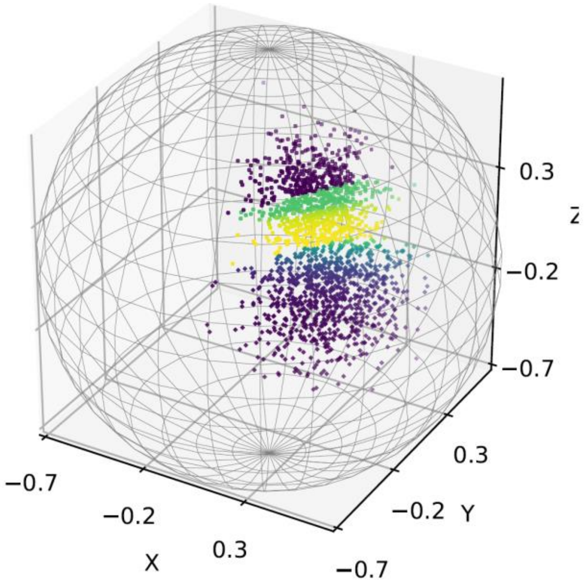


Theoretical Foundation:
Soudry et al, The Implicit Bias of Gradient Descent on Separable Data, ICLR18

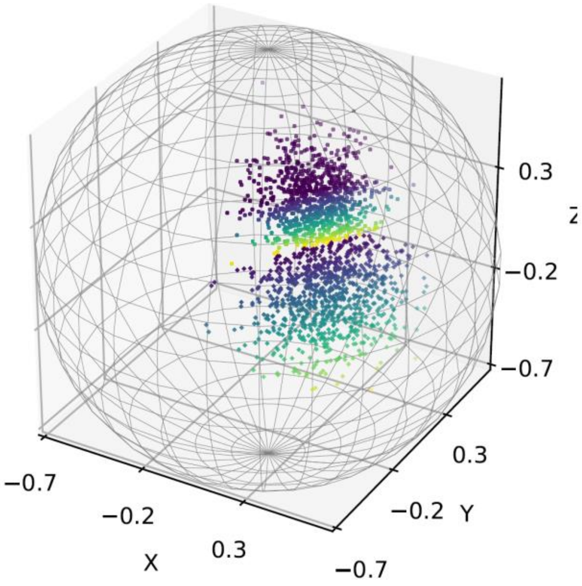
Toy Example: AVH vs. ||x||



Raw data

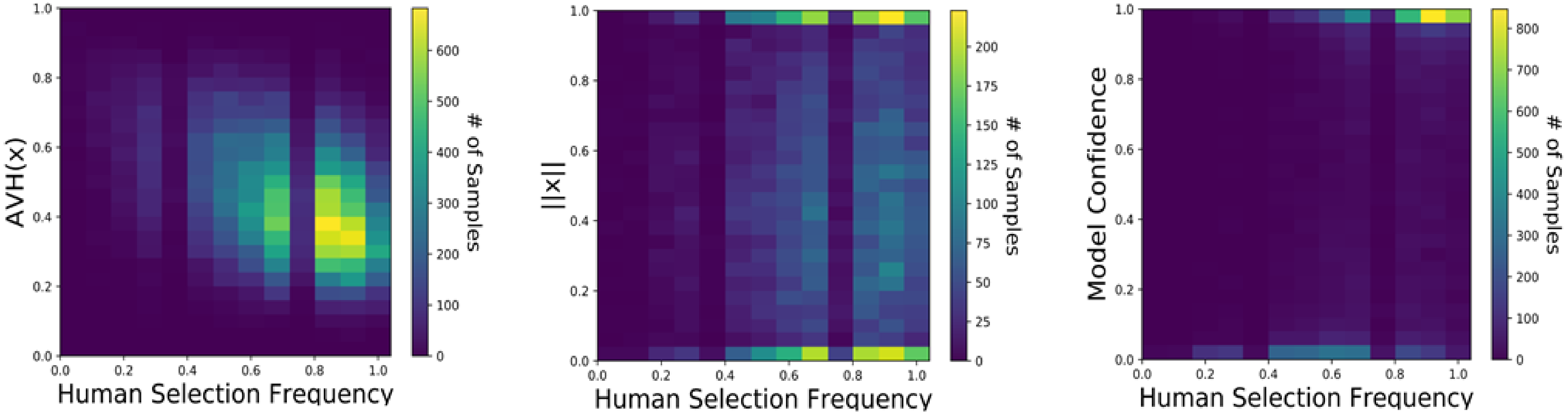


Heat map of AVH



Heat map of ||x||

Correlation between Different Measures and HSF

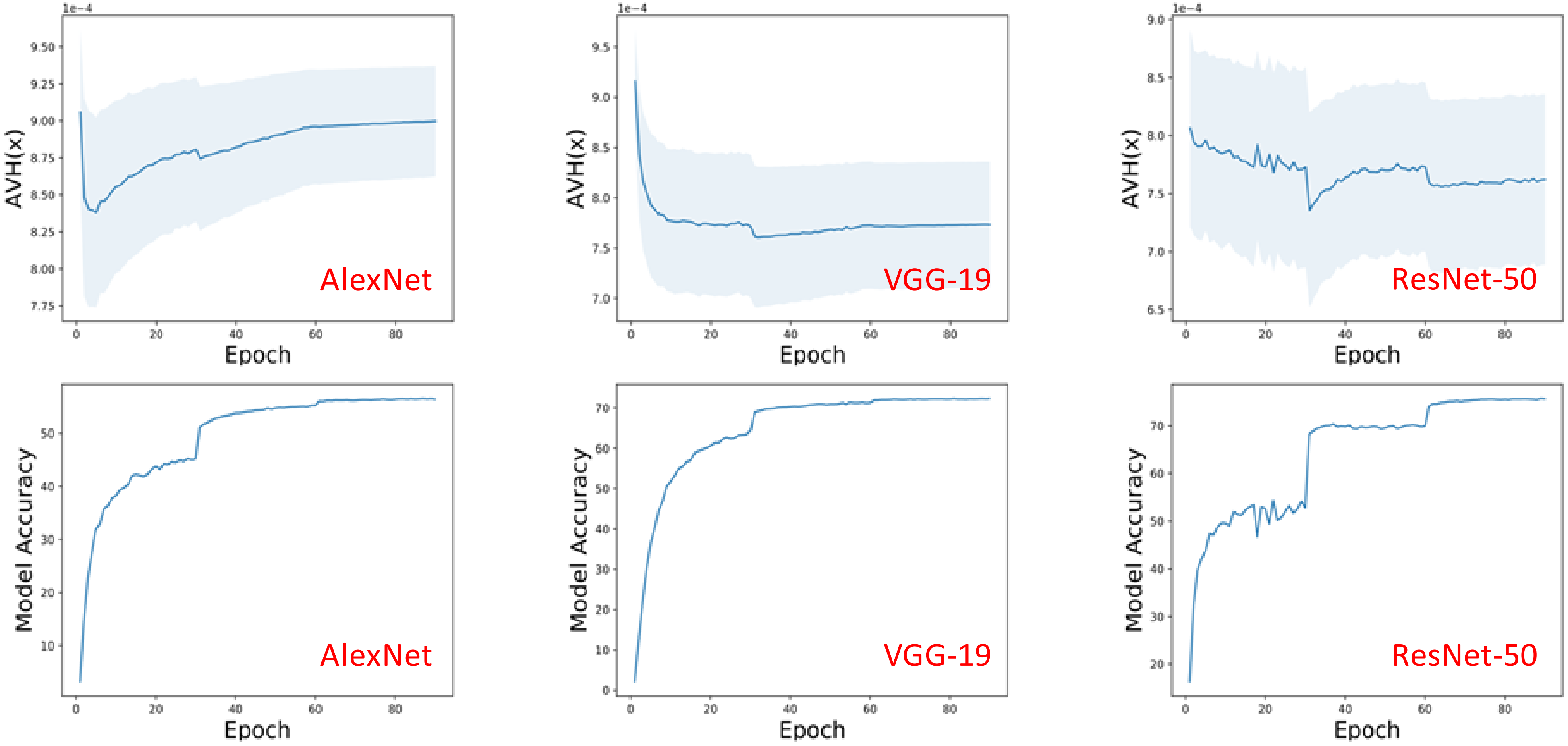


Spearman rank correlations

	z-score	Total Coef	$[0, 0.2]$	$[0.2, 0.4]$	$[0.4, 0.6]$	$[0.6, 0.8]$	$[0.8, 1.0]$
Number of Samples	-	29987	837	2732	6541	11066	8811
AVH	0.377	0.36	0.228	0.125	0.124	0.103	0.094
Model Confidence	0.337	0.325	0.192	0.122	0.102	0.078	0.056
$\|\mathbf{x}\|_2$	-	0.0017	0.0013	0.0007	0.0005	0.0004	0.0003

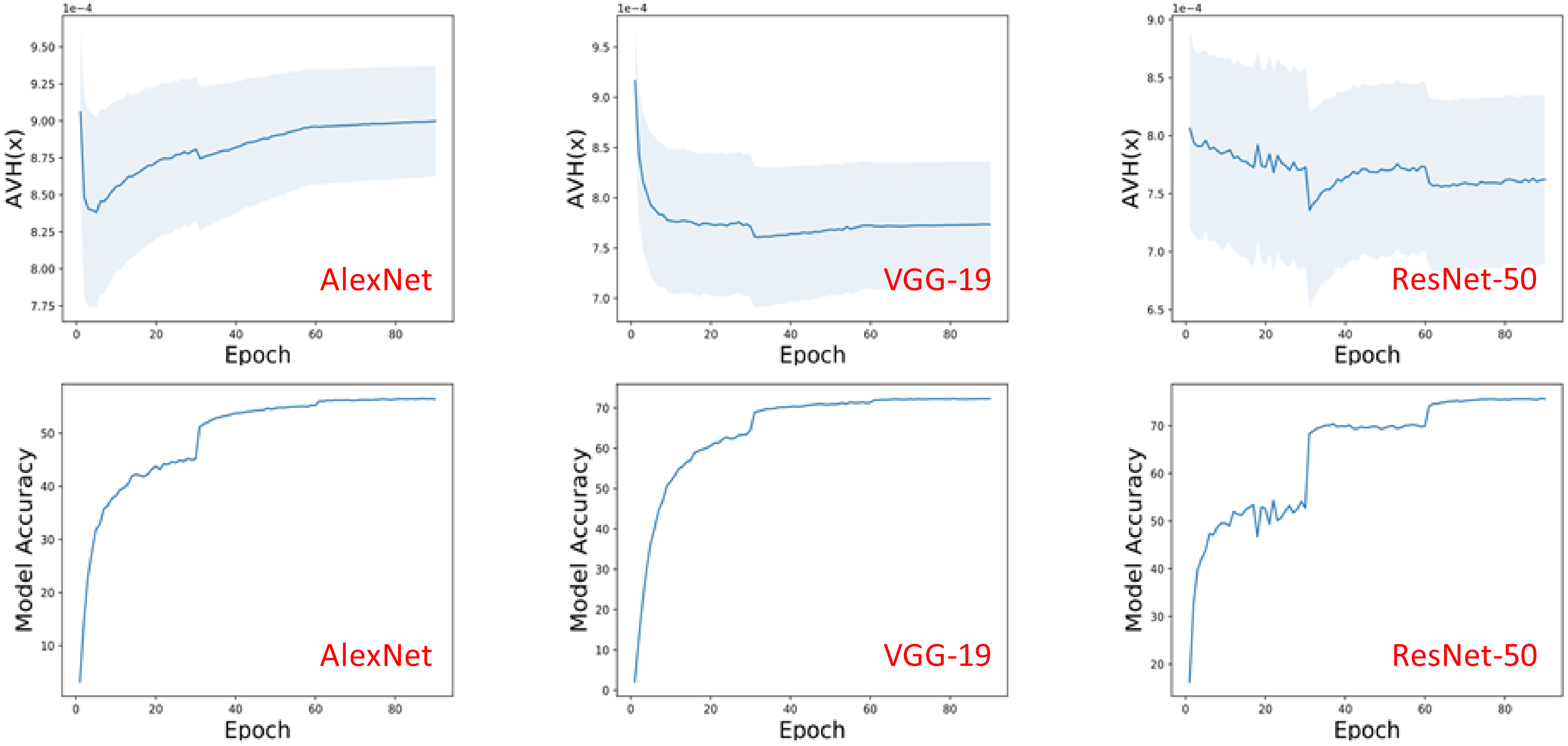
Main Discoveries

Discovery 1 - AVH hits plateau early even though accuracy or loss is still improving



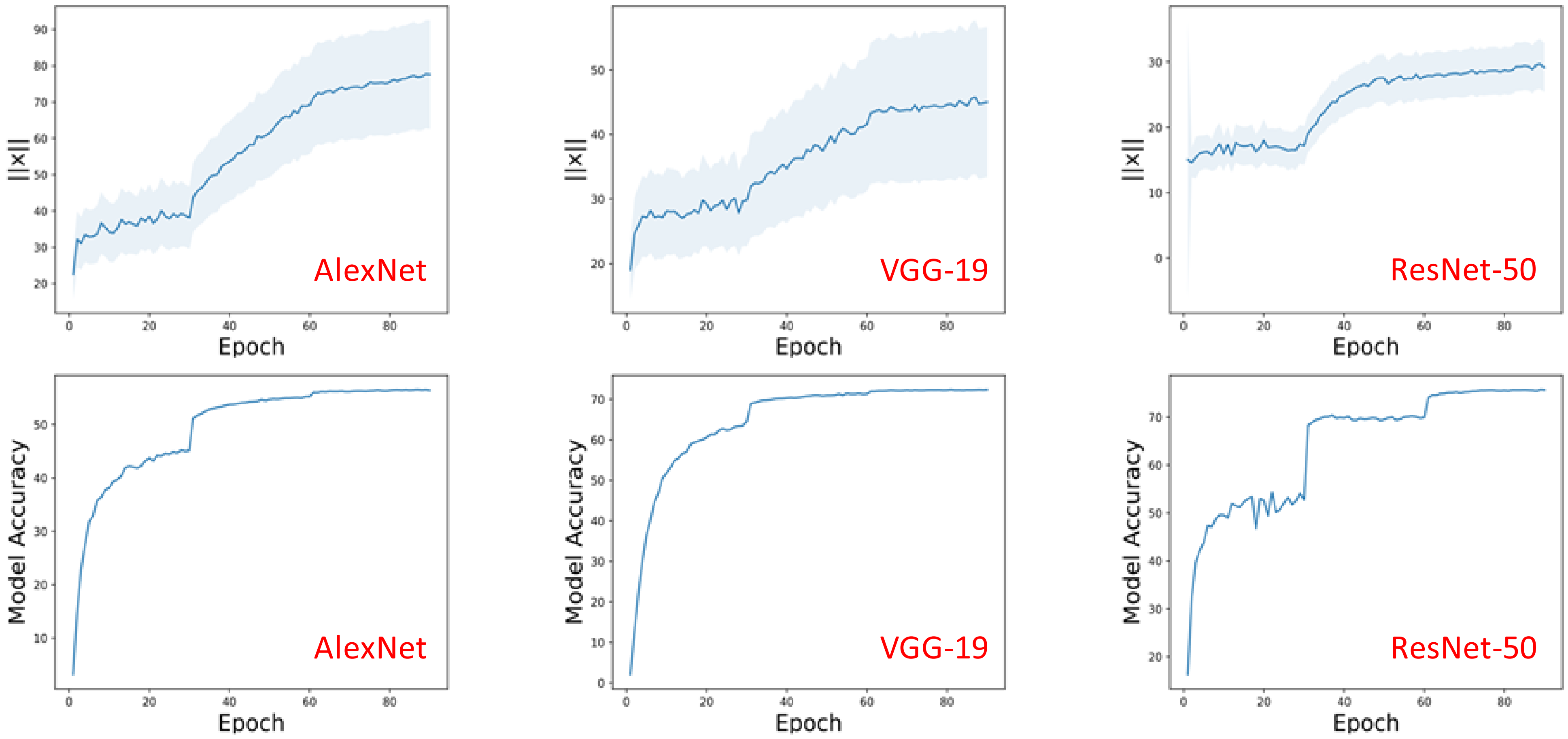
Main Discoveries

Discovery 2 - AVH is an indicator of model's generalization ability



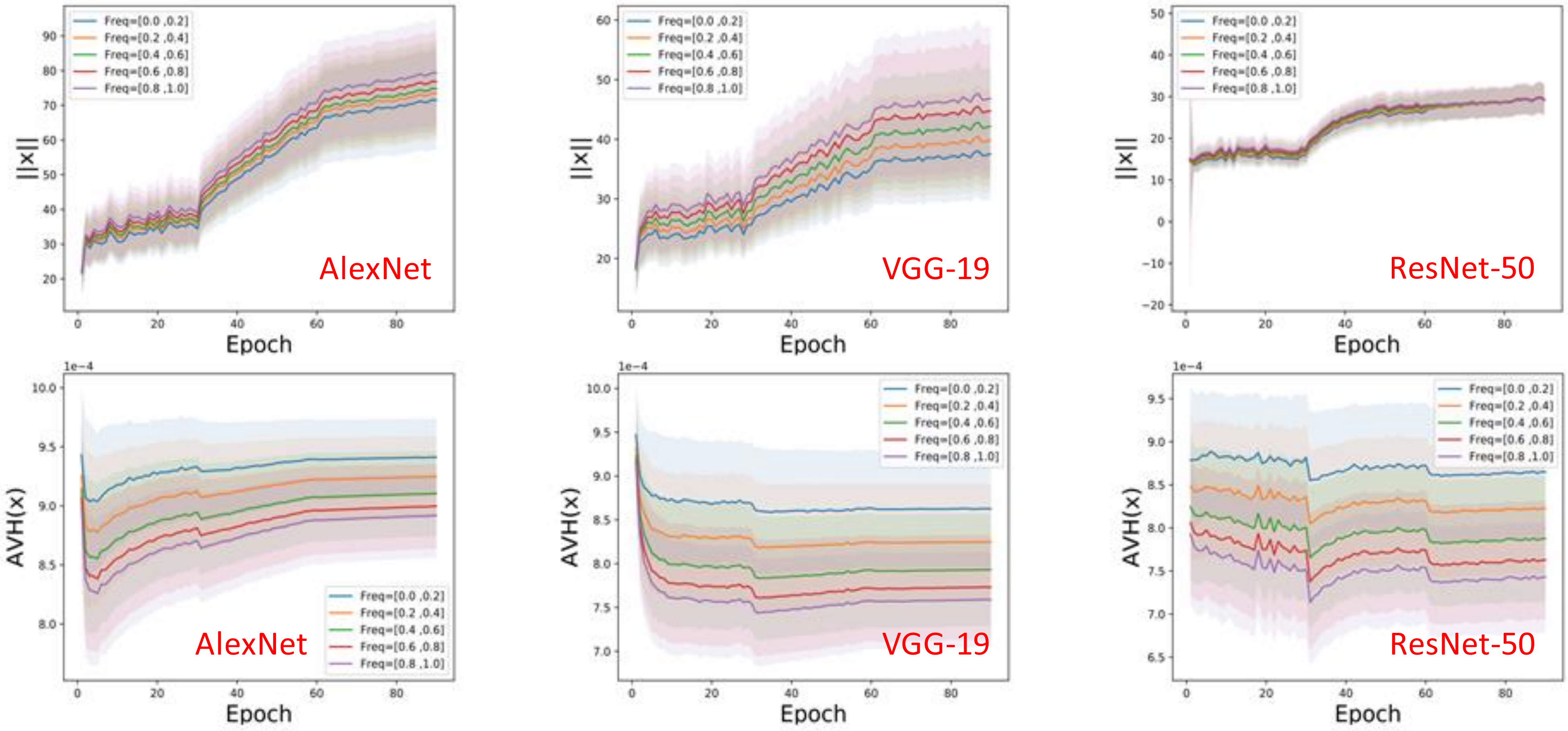
Main Discoveries

Discovery 3 - The norm of feature embeddings keeps increasing during training



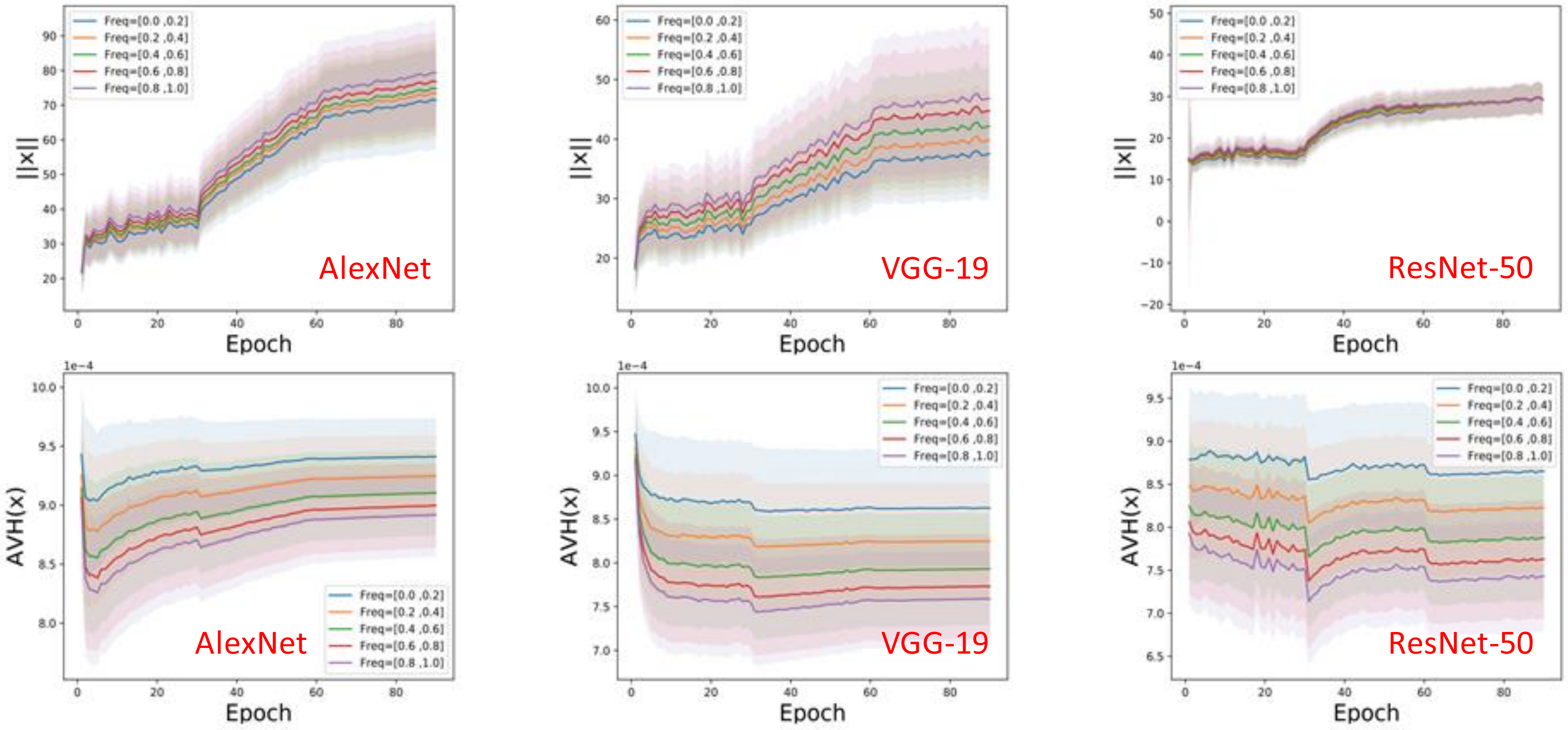
Main Discoveries

Discovery 4 - Correlation between AVH and human selection freq holds across models



Main Discoveries

Discovery 5 - Correlation between norm and human selection frequency is not consistent



Conjecture on training dynamic of CNNs

- Softmax cross-entropy loss will first optimize the angles among different classes while the norm will fluctuate and increase very slowly.
- The angles become more stable and change very slowly while the norm increases rapidly.
- Easy examples: the angles get decreased enough for correct classification, the softmax cross-entropy loss can be well minimized by increasing the norm.
- Hard examples: the plateau is cause by unable to decrease the angle to correctly classify examples or increase the norms otherwise hurting loss.

Application I: Self-Training for Domain Adaptation

VisDA17 Dataset

Car

Source Domain (Labeled)

Adaptation

Target Domain (Unlabeled)

CBST

$$\hat{y}_t^{(k)*} = \begin{cases} 1, \text{if } k = \arg\max_{c}\{\frac{p(c|\mathbf{x}_t;\mathbf{w})}{\lambda_c}\} \\ \quad \text{and} \quad p(k|\mathbf{x}_t;\mathbf{w}) > \lambda_k \\ 0, \text{otherwise} \end{cases}$$

CBST + AVH

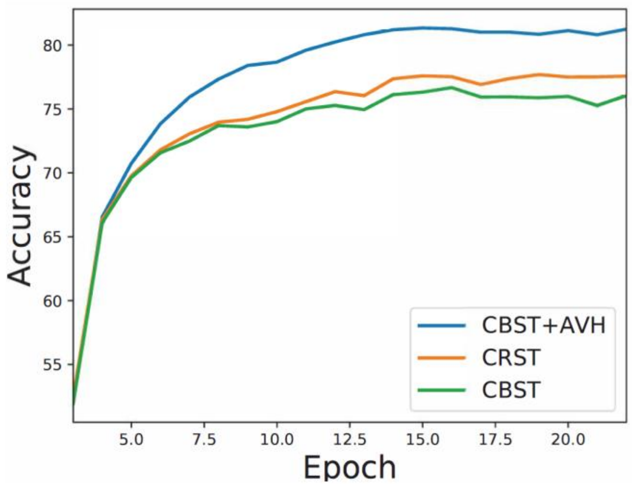
$$\mathcal{AVC}(c|\mathbf{x};\mathbf{w}) = \frac{\pi - \mathcal{A}(\mathbf{x},\mathbf{w}_c)}{\sum_{k=1}^{K}(\pi - \mathcal{A}(\mathbf{x},\mathbf{w}_k))}$$

$$\hat{y}_t^{(k)*} = \begin{cases} 1, \text{if } k = \arg\max_{c}\{\frac{p(c|\mathbf{x}_t;\mathbf{w})}{\lambda_c}\} \\ \quad \text{and } \mathcal{AVC}(k|\mathbf{x}_t;\mathbf{w}) > \beta_k \\ 0, \text{otherwise} \end{cases}$$

Improved selection

NVIDIA

Application I: Self-Training for Domain Adaptation

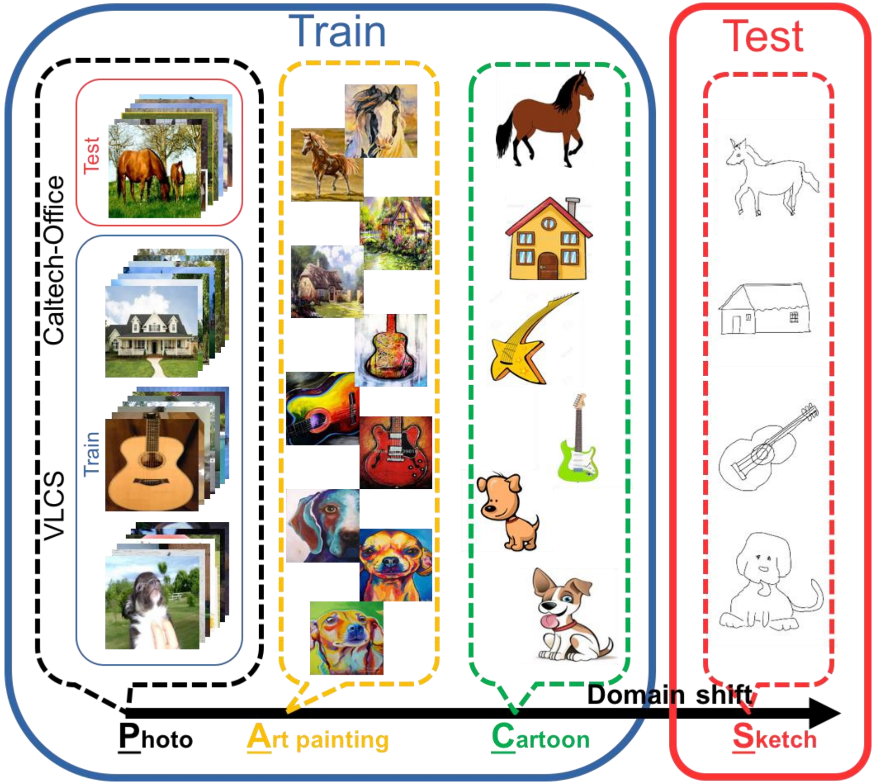


Examples chosen by AVH but not Softmax

Method	Aero	Bike	Bus	Car	Horse	Knife	Motor	Person	Plant	Skateboard	Train	Truck	Mean
Source (Saito et al., 2018)	55.1	53.3	61.9	59.1	80.6	17.9	79.7	31.2	81.0	26.5	73.5	8.5	52.4
MMD (Long et al., 2015b)	87.1	63.0	76.5	42.0	90.3	42.9	85.9	53.1	49.7	36.3	**85.8**	20.7	61.1
DANN (Ganin et al., 2016)	81.9	77.7	82.8	44.3	81.2	29.5	65.1	28.6	51.9	54.6	82.8	7.8	57.4
ENT (Grandvalet & Bengio, 2005)	80.3	75.5	75.8	48.3	77.9	27.3	69.7	40.2	46.5	46.6	79.3	16.0	57.0
MCD (Saito et al., 2017b)	87.0	60.9	**83.7**	64.0	88.9	79.6	84.7	76.9	88.6	40.3	83.0	25.8	71.9
ADR (Saito et al., 2018)	87.8	79.5	**83.7**	65.3	**92.3**	61.8	**88.9**	73.2	87.8	60.0	85.5	32.3	74.8
Source (Zou et al., 2019)	68.7	36.7	61.3	**70.4**	67.9	5.9	82.6	25.5	75.6	29.4	83.8	10.9	51.6
CBST (Zou et al., 2019)	87.2	78.8	56.5	55.4	85.1	79.2	83.8	77.7	82.8	**88.8**	69.0	**72.0**	76.4
CRST (Zou et al., 2019)	88.0	79.2	61.0	60.0	87.5	81.4	86.3	78.8	85.6	86.6	73.9	68.8	78.1
Proposed	**93.3**	**80.2**	78.9	60.9	88.4	**89.7**	**88.9**	**79.6**	**89.5**	86.8	81.5	60.0	**81.5**

Application II: AVH Loss for Domain Generalization

PACS Dataset



$$\mathcal{L}_{AVH} = \sum_i \frac{\exp\left(s(\pi - \mathcal{A}(\mathbf{x}_i, \mathbf{w}_{y_i}))\right)}{\sum_{k=1}^{K} \exp\left(s(\pi - \mathcal{A}(\mathbf{x}_i, \mathbf{w}_k))\right)}$$

Method	Painting	Cartoon	Photo	Sketch	Avg
AlexNet (Li et al., 2017)	62.86	66.97	89.50	57.51	69.21
MLDG (Li et al., 2018)	66.23	66.88	88.00	58.96	70.01
MetaReg (Balaji et al., 2018)	**69.82**	70.35	**91.07**	59.26	**72.62**
Feature-critic (Li et al., 2019)	64.89	**71.72**	89.94	**61.85**	72.10
Baseline CNN-9	66.46	67.88	89.70	51.72	68.94
CNN-9 + AVH	**71.56**	**69.25**	**89.93**	**60.86**	**72.90**

Thanks You!