Towards understanding transfer learning and its limits

Hanie Sedghi

Google Research, Brain Team

Science of Deep Learning

- Our understanding of modern neural networks lags behind their practical successes. This growing gap poses a challenge to the pace of progress.
- Although there has been some progress in this area, still we are far from answering many fundamental questions such as generalization capabilities of deep models and how to ensure successful transfer to new domains.
- I believe this understanding helps us extend beyond our current use of deep learning in a reliable way.

Science of Deep Learning

- Our understanding of modern neural networks lags behind their practical successes. This growing gap poses a challenge to the pace of progress.
- Although there has been some progress in this area, still we are far from answering many fundamental questions such as generalization capabilities of deep models and how to ensure successful transfer to new domains.
- I believe this understanding helps us extend beyond our current use of deep learning in a reliable way.
- Principled approaches to investigate deep learning phenomena.
- To understand when and why DNNs generalize, improve training and generalization performance in state of the art deep learning models and extend the current success of our models to new domains.

Part I

What is being transferred in transfer learning?

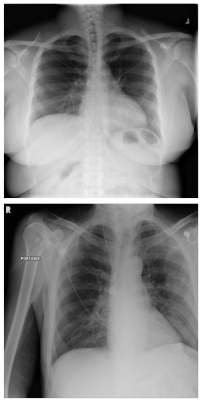
What is being transferred in transfer learning?

- One desired capability of machines is to transfer their knowledge or understanding of a domain it is trained on (source domain) to another domain (target domain) where data is (usually) scarce or a fast speed of convergence is needed.
- Plethora of works using transfer learning in different applications.
- We would like to understand:
  - what enables a successful transfer?
  - which parts of the network are responsible for that?

What is being transferred in transfer learning?, B. Neyshabur*, H. Sedghi*, C. Zhang* , NeurIPS 2020

Problem Setup

- Target domains that are intrinsically different and diverse:
  - CheXpert: a medical imaging dataset of chest x-rays considering 5 different diseases.
  - DomainNet: designed to probe transfer learning for diverse visual representations. The domains range from real images to sketches, clipart and painting samples. 345 classes
- Two initialization scenarios:
  - Pre-trained on ImageNet (Finetune)
  - Start from random initialization (RandInit)

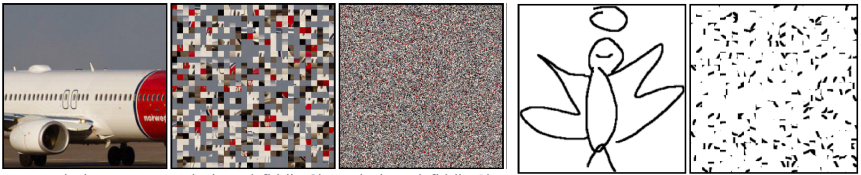


Role of feature reuse

- Comparing learning curves
  - Largest performance boost on the `real` domain, which contains natural images.
  - Even for the most distant target domains, we still observe performance boosts from transfer learning.
  - The optimization for Finetune also converges much faster than Randinit in all cases.
- The benefits of transfer learning are generally believed to come from reusing the pre-trained feature hierarchy.
- But, why in many successful applications of transfer learning, the target domain could be visually very dissimilar to the source domain?

Role of feature reuse

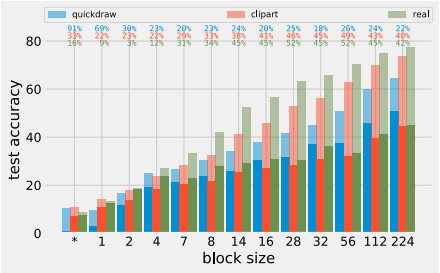
Experiment: We partition the image of the downstream tasks into equal sized blocks and shuffle the blocks randomly. The shuffling disrupts visual features in those images.



Role of feature reuse

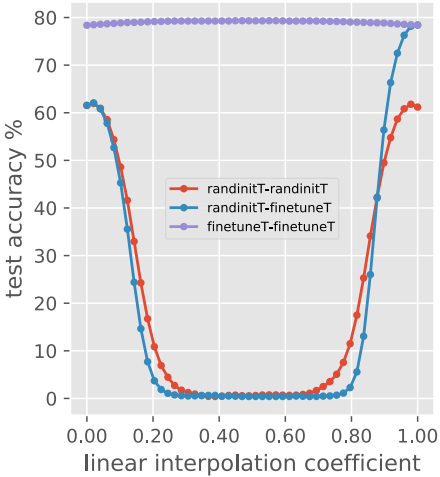
- Feature reuse plays a very important role!
  especially when the downstream task shares similar visual features with the pre-training domain.

- There are other factors at play!
  low-level statistics of the data that are not ruined in the shuffling lead to the significant benefits of transfer learning, especially on optimization speed.



Performance barriers in the loss landscape

- *Any* two minimizers of a deep network can be connected via a *non-linear* low-loss path.
- We evaluate a series of models along the linear interpolation of the two weights.
- Performance barriers are generally expected between two unrelated NN models.
- When the two solutions belong to the same flat basin of the loss landscape, performance barrier is absent.
- Finetune models reside in the same basin.
- RandInits end up in a different basin, even if starting from same random seed.



Performance barriers in the loss landscape

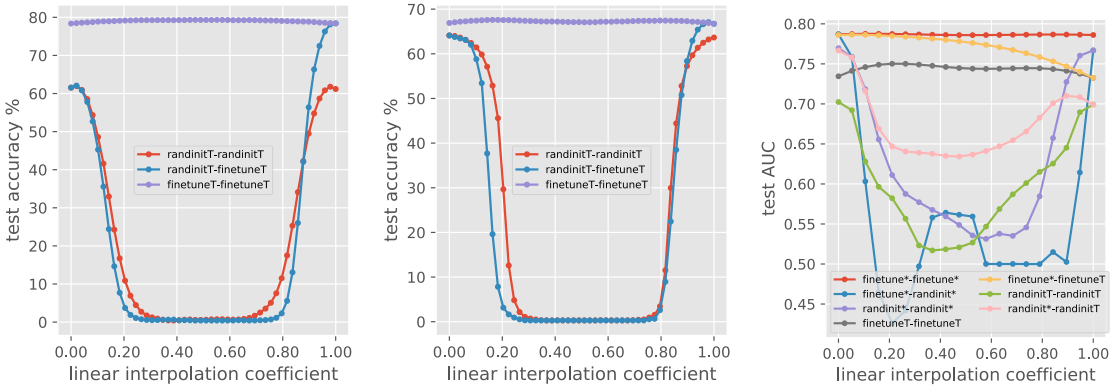
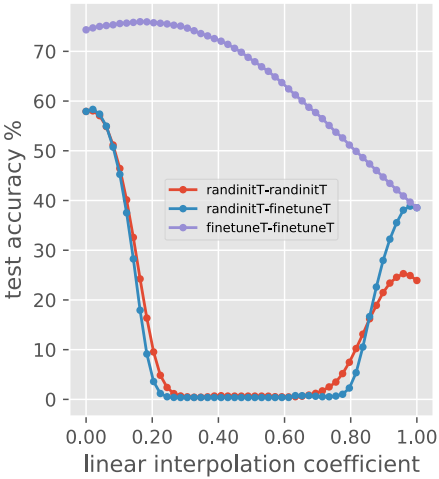


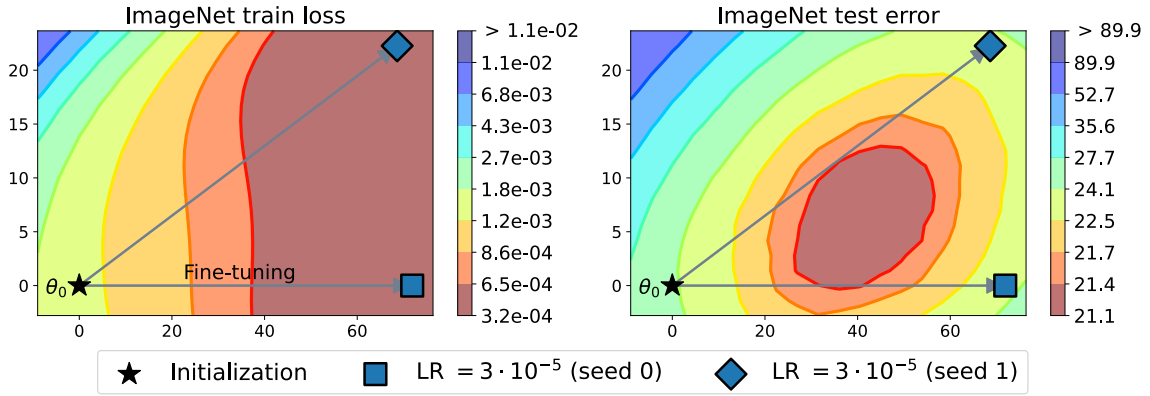
Figure: The left and middle panes show performance barrier measured by test accuracy on DomainNet `real` and `quickdraw`, respectively. The right pane shows the performance barrier measured by test AUC on CheXpert.

Cross-domain weight interpolation on DomainNet

- when directly evaluated on a different domain that the models are trained from, we could still get non-trivial test performance.
- P-T consistently outperforms RI-T even in the cross-domain cases.
- when interpolating between P-T models, (instead of performance barrier) we observe performance boost in the middle of the interpolation.
- This suggests that all the trained P-T models on all domains are in one shared basin.



Model Soups



Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time, Wortsman et al 2022

Model Soups

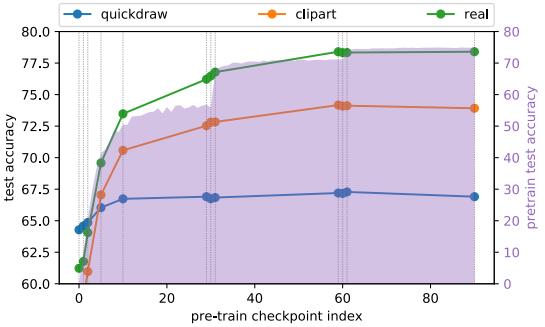
Method	ImageNet			Distribution shifts					
	Top-1	ReaL	Multilabel	IN-V2	IN-R	IN-Sketch	ObjectNet	IN-A	Avg shifts
ViT/G-14 (Zhai et al., 2021)	90.45	90.81	–	83.33	–	–	70.53	–	–
CoAtNet-7 (Dai et al., 2021)	90.88	–	–	–	–	–	–	–	–
Our models/evaluations based on ViT-G/14:									
ViT/G-14 (Zhai et al., 2021) (reevaluated)	90.47	90.86	96.89	83.39	94.38	72.37	71.16	89.00	82.06
Best model on held out val set	90.72	91.04	96.94	83.76	95.04	73.16	78.20	91.75	84.38
Best model on each test set (oracle)	90.78	**91.78**	**97.29**	**84.31**	95.04	73.73	**79.03**	92.16	84.68
Greedy ensemble	**90.93**	91.29	**97.23**	**84.14**	94.85	73.07	77.87	91.69	84.33
Greedy soup	**90.94**	91.20	**97.17**	**84.22**	**95.46**	**74.23**	78.52	**92.67**	**85.02**

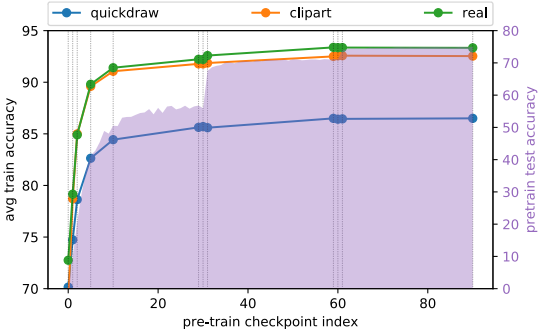
- No barrier between different fune-tuned model $\rightarrow$ possible to combine fine-tuned models by interpolating their weights.
- Simply averaging the weights of multiple models fine-tuned with different hyperparameters can improve performance
- Achieving most of the accuracy gain of ensembling outputs without any added computational cost at inference time.

Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time, Wortsman et al 2022

Which pre-trained checkpoint is most useful for transfer learning?

- Significant improvements are observed when we start from the checkpoints where the pre-training performance has been plateauing.
- Independence between the improvements on optimization speed and final performance.
- You can start from earlier checkpoints in pre-training.





Not all layers are created equal!

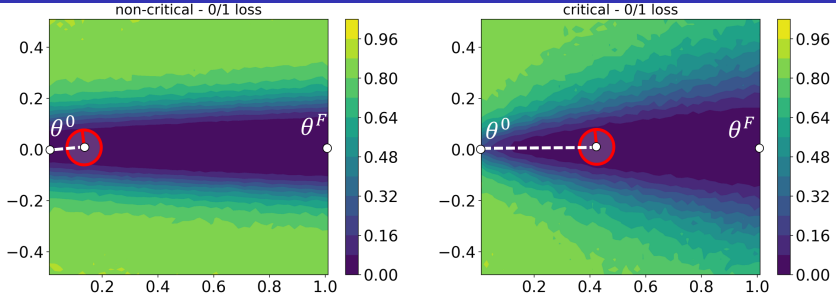
Experiment:

- Consider a deep neural network (at any training epoch).
- Pick one of the layers and rewind its value back to its value at initialization.
- Keep the value of all other layers fixed.
- Notice the change in performance.

Observation: In a deep neural network, some modules are more critical than others, i.e., rewinding their parameter values back to initialization, while keeping other modules fixed at the trained parameters, results in a large drop in the network's performance.

C. Zhang, S. Bengio, Y. Singer, *Are all layers created equal?*, Feb 2019

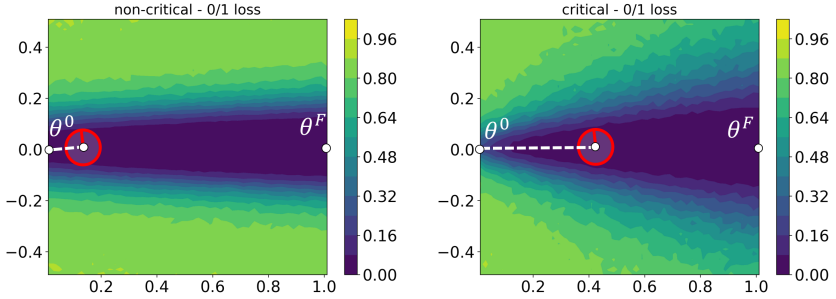
Role of different layers: Module Criticality



- Loss values in the valleys that connect the initial weights θ^0 to the final weights θ^F.
- Module criticality: how far one can push the ball of radius r in the valley towards initialization divided by the radius.

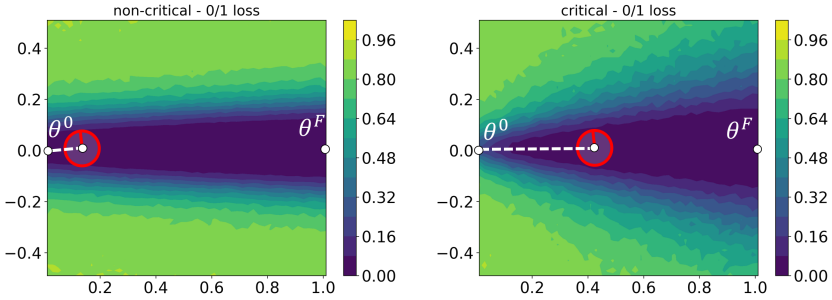
The intriguing role of module criticality in the generalization of deep networks,
N. Chatterji, B. Neyshabur, H. Sedghi, Spotlight in ICLR2020

Module Criticality



- Non-critical modules $\equiv$ wide valley
  Critical modules $\equiv$ sharp valley

Module Criticality

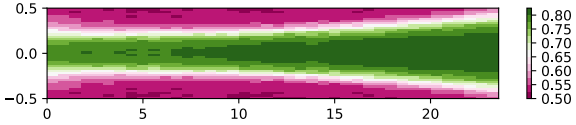


- Non-critical modules $\equiv$ wide valley
  Critical modules $\equiv$ sharp valley
- Module criticality as a generalization measure correlates well with model performance.

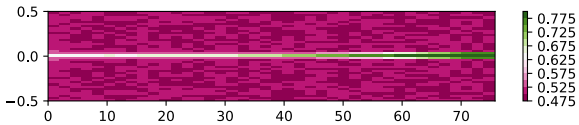
The intriguing role of module criticality in the generalization of deep networks,
N. Chatterji, B. Neyshabur, H. Sedghi, Spotlight in ICLR2020

Role of different layers

- As we move from the input towards the output, we see tighter valleys, i.e., modules become more critical.
- This is in agreement with observation of [Yosinski+2014, Raghu+2019] that lower layers are in charge of more general features while higher layers have features that are more specialized for the target domain.



(g) Module Criticality Layer1



(h) Module Criticality Layer4

So far...

- Both feature-reuse and low-level statistics of the data are important.
- There is no performance barriers between finetune models, while models trained from random initialization are in a different basins in the loss landscape.
- Lower layers are in charge of general features and higher layers are more sensitive to perturbation of their parameters.

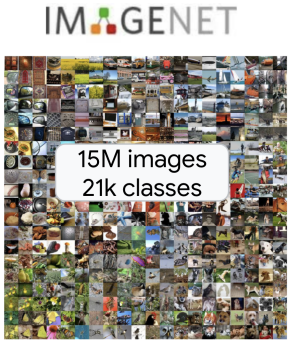
What is being transferred in transfer learning?, B. Neyshabur*, H. Sedghi*, C. Zhang* , NeurIPS 2020

Part II

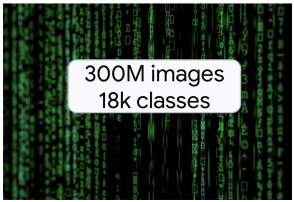
Exploring the limits of large scale pre-training

Effect of Scale: A prelude

- Recent impressive progress on transfer and few-shot learning: scaling up model and data
- Prominent examples: GPT-3, CLIP
- Massive datasets: Instagram images and JFT-300



"JFT-300M"
Google internal dataset



Effect of scale: current narrative

- These developments implicitly encourage two consistent views:
  1. Scaling up the model and data size improves the performance significantly;
  2. The performance improvement transfers to downstream tasks in a desirable way.
- Non-saturating performance.
- Linear relationship between imagenet pre-training and downstream accuracy.

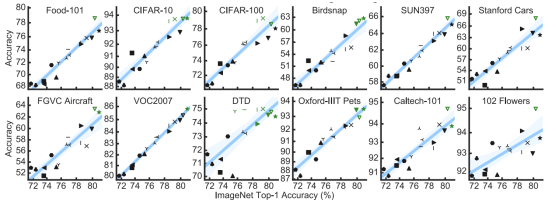


Figure: Kornblith et al, 2019

Shortcomings of earlier works

- Performance for different choices of hyper-parameter values are not reported.
- When studying scaling, we are concerned about the best performance of models given all possible values for the hyper-parameters!
- Limited accuracy range

Shortcomings of earlier works

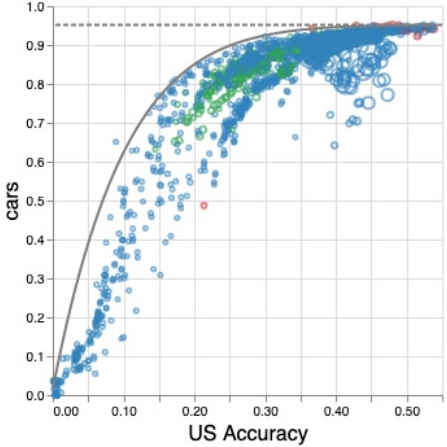
- Performance for different choices of hyper-parameter values are not reported.
- When studying scaling, we are concerned about the best performance of models given all possible values for the hyper-parameters!
- Limited accuracy range
- Focusing on improving SOTA and limited computational budget.
- Simply extrapolating scaling without understanding of the dynamics of scaling can be detrimental.

This work

- Systematic large scale study
- Investigate the transferability of improvements on a large-scale upstream task to a wide range of downstream tasks.
- More than 4800 experiments
- Image recognition task
- Vision Transformers, ResNets, Mixers of varying size (ten million to ten billion parameters)
- Trained on the largest scale of available image data (JFT, ImageNet21k)
- More than 20 downstream tasks
- Downstream tasks cover a wide range of standard datasets, e.g., VTAB, MetaDataset, Wilds and medical imaging.

Setting

- Goal: Predict downstream performance for a given model.
- DS-vs-US accuracy plot.
- Horizontal line = Predicted accuracy as US accuracy becomes 1.



Recap: Convex hull

Definition (Convex hull)

A convex hull $\mathcal{C}$ of N points a_j in a set $\mathcal{S}$ is given by

$$\mathcal{C} \equiv \left\{ \sum_{j=1}^{N} p_j a_j : p_j \geq 0 \quad \text{for all j,} \quad \sum_{j=1}^{N} p_j = 1 \right\}.$$

Recap: Convex hull

Definition (Convex hull)

A convex hull $\mathcal{C}$ of N points a_j in a set $\mathcal{S}$ is given by

$$\mathcal{C} \equiv \left\{ \sum_{j=1}^{N} p_j a_j : p_j \geq 0 \text{ for all j, } \sum_{j=1}^{N} p_j = 1 \right\}.$$

Lemma

Consider a group of models $\theta_j, j \in [N]$ that reaches accuracy $a_j = (a_j^{US}, a_j^{DS}), j \in [N]$ on some pair of tasks (US,DS). Construct a randomized model $\tilde{\theta}$ as follows: for each input x_i, with probability p_j pick model θ_j and output $\theta_j(x_i)$. Then the randomized model will demonstrate accuracy $\sum_{j=1}^{N} p_j a_j$.

Choice of data for fitting the power law

Lemma

Consider a group of models $\theta_j, j \in [N]$ that reaches accuracy $a_j = (a_j^{US}, a_j^{DS}), j \in [N]$ on some pair of tasks (US,DS). Construct a randomized model $\tilde{\theta}$ as follows: for each input x_i, with probability p_j pick model θ_j and output $\theta_j(x_i)$. Then the randomized model will demonstrate accuracy $\sum_{j=1}^{N} p_j a_j$.

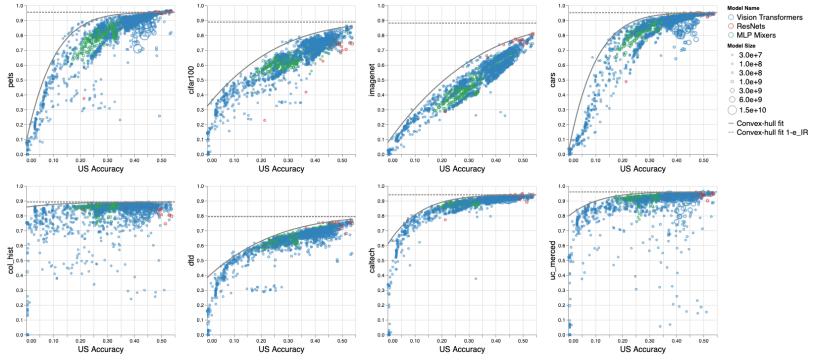
$\Rightarrow$ We consider the convex hull of the points in our analysis.

Large variance in DS-vs-US performance across models.

1. fit the existing points $\rightarrow$ fit average performance
2. fit the convex hull $\rightarrow$ fit best performing model. Robust to density of the points.

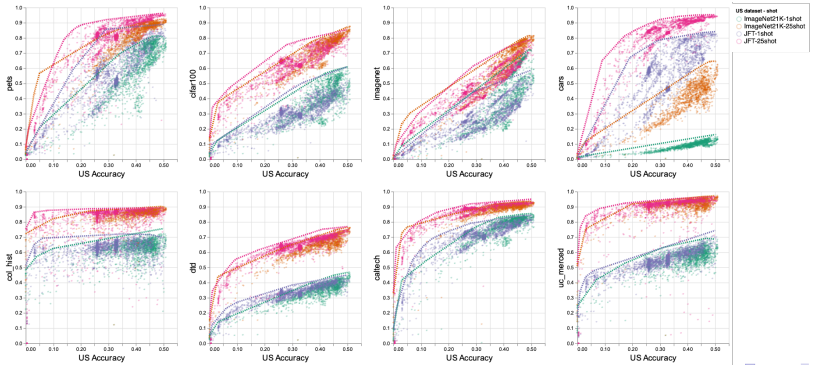
The diminishing benefit of scaling up in transfer learning

- Goal: Predict downstream performance for a given model.
- DS-vs-US accuracy plot.
- Saturation: even if US reaches accuracy of one, DS won't.
- Nonlinear relationship.



The diminishing benefit of scaling up in transfer learning

- DS-vs-US accuracy plot.
- Saturation
- Nonlinear relationship.
- Consistent across different US tasks & No. of shots.

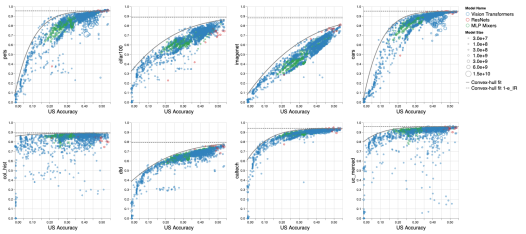


Scaling laws for downstream accuracy

- Goal: Predict DS performance
- Our proposed model

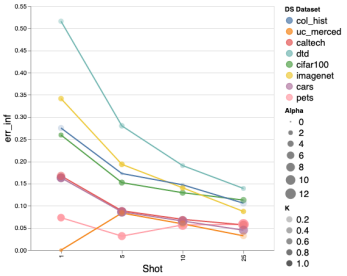
$$e_{DS} = k(e_{US})^{\alpha} + e_{\infty}$$

- e_{∞}
  - irreducible error.
  - captures the value of DS error if US error reaches zero.
  - captures the nonlinearity.
  - is not the Bayes error.



Effect of design choices on power law parameters

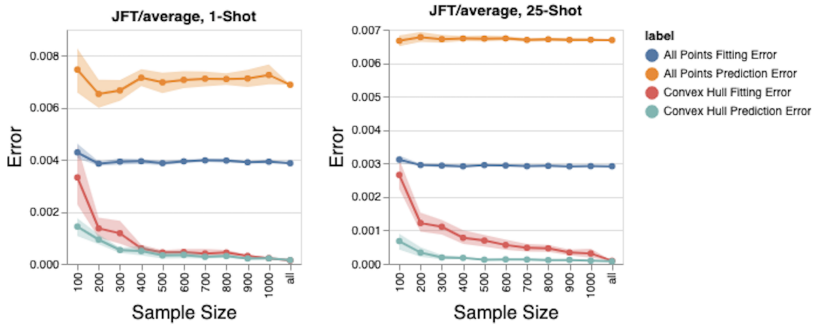
- k, e_∞ correlate negatively with number of shots.
- α is positively correlated with number of shots.
- Correlation values change drastically for different choices of US, DS tasks.



DS	US	Parameter	Correlation with Number of Shots
caltech	ImageNet21K	K	-0.777892
caltech	ImageNet21K	α	-0.582066
caltech	ImageNet21K	e_∞	-0.845368
caltech	JFT	K	-0.620526
caltech	JFT	α	0.259305
caltech	JFT	e_∞	-0.762856
cars	ImageNet21K	K	0.720391
cars	ImageNet21K	α	0.960490
cars	ImageNet21K	e_∞	-0.737273
cars	JFT	K	-0.976599
cars	JFT	α	-0.034033
cars	JFT	e_∞	-0.809016
cifar100	ImageNet21K	K	-0.918914
cifar100	ImageNet21K	α	0.683485
cifar100	ImageNet21K	e_∞	-0.587304
cifar100	JFT	K	-0.934455
cifar100	JFT	α	0.707966
cifar100	JFT	e_∞	-0.754030

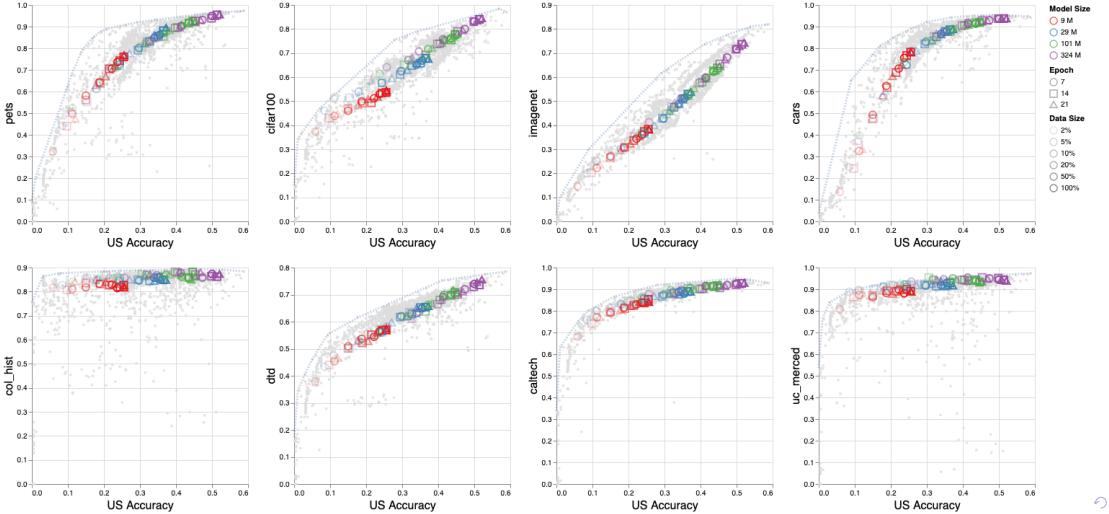
Sample size sensitivity analysis

- The prediction error is very small across all these choices.
- The proposed model will work well even when we have much smaller number of DS-vs-US points.
- The fitting error decreases by increasing the number of samples.



Effect of scale: A closer look

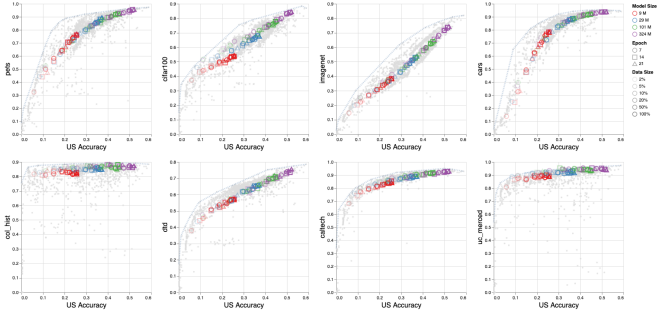
Controlled experiments : model size, data size, compute



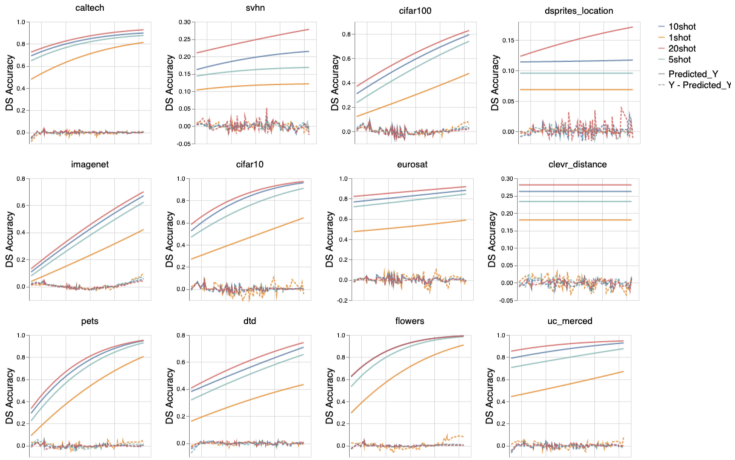
Effect of scale: A closer look

Controlled experiments : model size, data size, compute

- Same pattern.
- Same curve for the 3 parameters.
- Grid search equivalence.
- Variation is due to training hyper-parameters.



On the prediction power of US accuracy



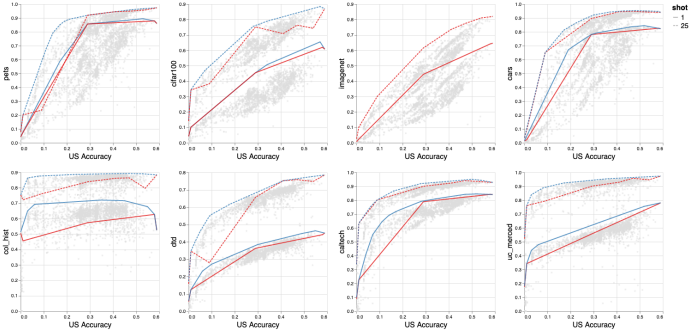
DS	error std
birds	0.1542
caltech	0.1020
cars	0.1979
col_hist	0.1552
dtd	0.0885
imagenet	0.1882
pets	0.1412
uc_merced	0.1581

Conditioned on US accuracy, not much is left for the rest of parameters altogether to predict!

Investigating different DS-vs-US trends

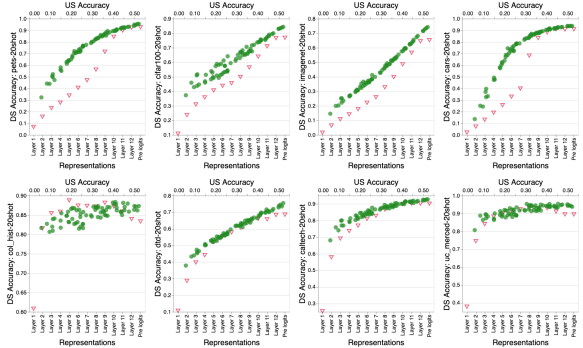
Overlay the convex hull of ImageNet DS-vs-US plot on all DS tasks. Observation

1. Best performing ImageNet models perform very similarly to best performing models in several but not all DS tasks.
2. As the US performance increases, the gap between best performing ImageNet models and best performing DS task models reduces significantly.



Investigating different DS-vs-US trends: Experiment

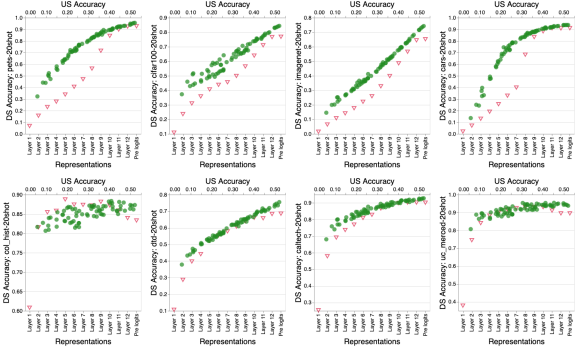
Experiment: Move the head to different layers



- DS versus US performance
- DS performance for representation taken from specific layer

Investigating different DS-vs-US trends: Experiment

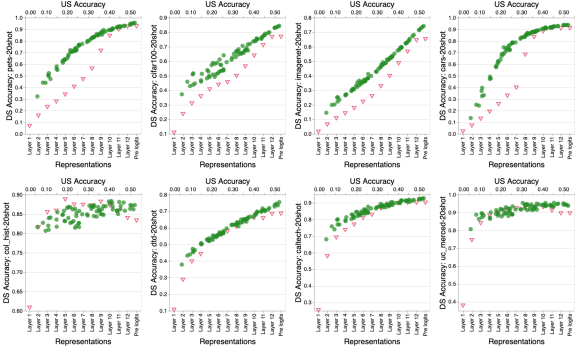
Experiment: Move the head to different layers



- DS versus US performance
- DS performance for representation taken from specific layer
- Plots show similar trend.
- For DS that saturate faster, higher layers are not needed.
- Lower layers capture lower level features that are more common across different dataset and tasks, whereas fine-grained features reside at top layers in the network

Investigating different DS-vs-US trends: Experiment

Experiment: Move the head to different layers

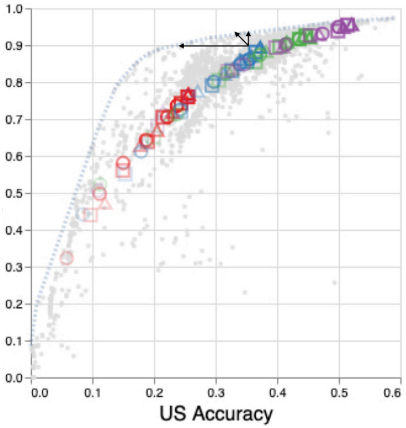


- DS versus US performance
- DS performance for representation taken from specific layer
- Plots show similar trend.
- For DS that saturate faster, higher layers are not needed.
- Lower layers capture lower level features that are more common across different dataset and tasks, whereas fine-grained features reside at top layers in the network

We need data diversity.

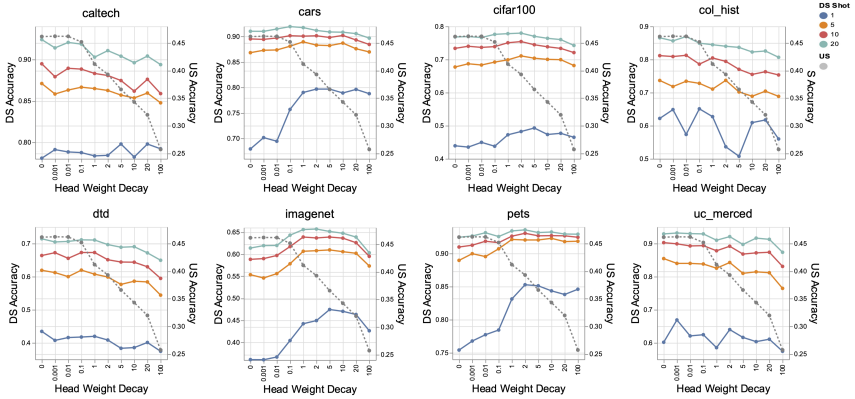
Discrepancies between US and DS performances: a case study

- Recap:
  training hyper-parameters cause variation from the curve.
- Now:
  focus on effect of head hyper-parameters.
- Decouple head from rest of network.
- weight decay, learning rate.



Effect of head weight decay

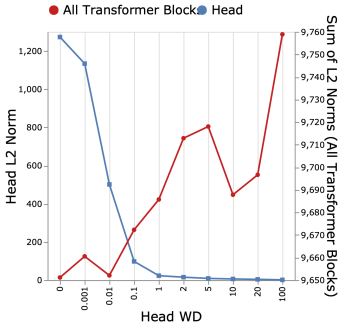
- Increasing head WD hurts US performance.
- Increasing head WD improves DS performance for some tasks.

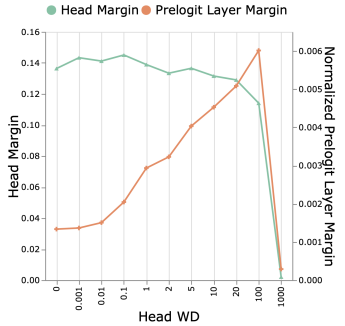


Discrepancies between US and DS performances: why?

Increasing head WD

- decreases head margin, increases layer margin [Elsayed et al 2018].
- decreases head norm, increases layer norm for lower layers.
- pushes the information down to lower layers.





On generalization of the observed phenomena

- Number of shots
- Transfer vs. few-shot
- Scaling of plots: logit$(p) = \log(\frac{p}{1-p})$, $-\log(1-p)$, linear
- Architecture

Summary

- Large-scale systematic study.
- Performance saturation on DS does happen.
- Modeled DS-vs-US accuracy and predict DS accuracy by a power law curve.
- Our model predicts saturation point and is robust to low sample size.
- Data diversity matters.
- Scaling model size, pre-training data size, compute leads to the same curve.
- US performance has high prediction power.
- Hyper-parameters used in training matter and need to be DS-specific.
- Head hyper-parameters are important and can help improve DS performance.

Zooming in on the role of data

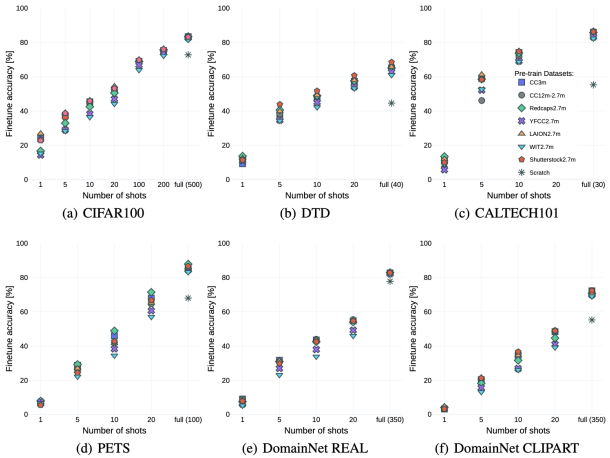
- Pre-training: CLIP, SimCLR
- Architecture: ResNet50
- 4000 trained networks.
- 7 upstream, 9 downstream datasets
- Downstream: CIFAR100, DTD, CALTECH101, PETS, Domainnet REAL, Domainnet CLIPART CameraTraps, Cassava Leaf Disease, EuroSAT

Dataset	Source	Total size
YFCC	Flickr	14,826,000
LAION	Common Crawl	15,504,742
CC-12M	Unspecified web pages	9,594,338
RedCaps	Reddit	11,882,403
WIT	Wikipedia	5,038,295
ShutterStock	ShutterStock	11,800,000
IN1K-Captions	ImageNet	463,622

The role of pretraining data in transfer learning, R. Entezari, M. Wortsman, O. Saukh, M. Shariatnia, H. Sedghi, Ludwig Schmidt, In submission

Role of pretraining data distribution

- CLIP
- the number of pretraining images is 2.7 million.
- Shutterstock is the best performing pre-training datasets
- pre-training dataset is important for low-shot transfer.

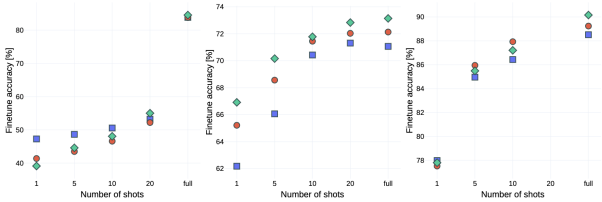


(a) CIFAR100 (b) DTD (c) CALTECH101

(d) PETS (e) DomainNet REAL (f) DomainNet CLIPART

Role of pretraining data distribution

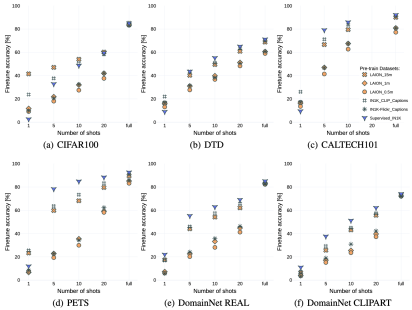
- Same setting as before
- Only change pretraining method to SimCLR
- Observe similar phenomena



(a) CIFAR100 (b) DTD (c) CALTECH101

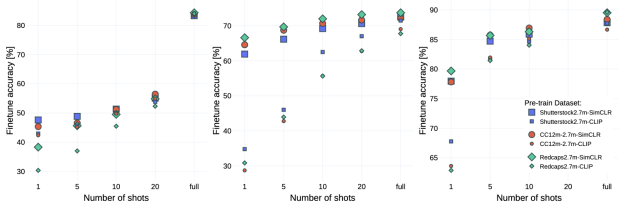
Role of data curation

- Recap:
  training hyper-parameters cause variation from the curve.



Role of pretraining method

- Contrastive > supervised for low shot setting
- image-image contrastive > image-text contrastive

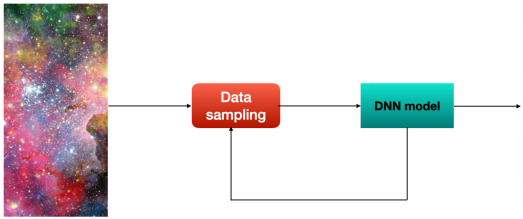


(a) CIFAR100 (b) DTD (c) CALTECH101

What's next?

Data sampling module

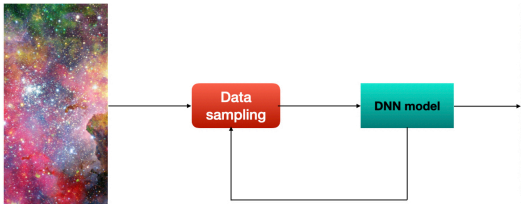
- Modeling data
- Ensuring data diversity
- Closing the loop
- Investigating the effect of curriculum learning



What's next?

Data sampling module

- Modeling data
- Ensuring data diversity
- Closing the loop
- Investigating the effect of curriculum learning



Thank you!