Object Classification

Lecture 4

Deep Residual Networks

Deep Learning Gets Way Deeper

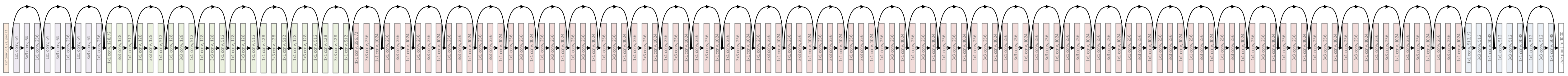
8:30-10:30am, June 19

ICML 2016 tutorial

Kaiming He

Facebook AI Research*

*as of July 2016. Formerly affiliated with Microsoft Research Asia



Introduction

Introduction

Deep Residual Networks (ResNets)

- "Deep Residual Learning for Image Recognition". CVPR 2016
- A simple and clean framework of training "very" deep nets

- State-of-the-art performance for
  - Image classification
  - Object detection
  - Semantic segmentation
  - and more…

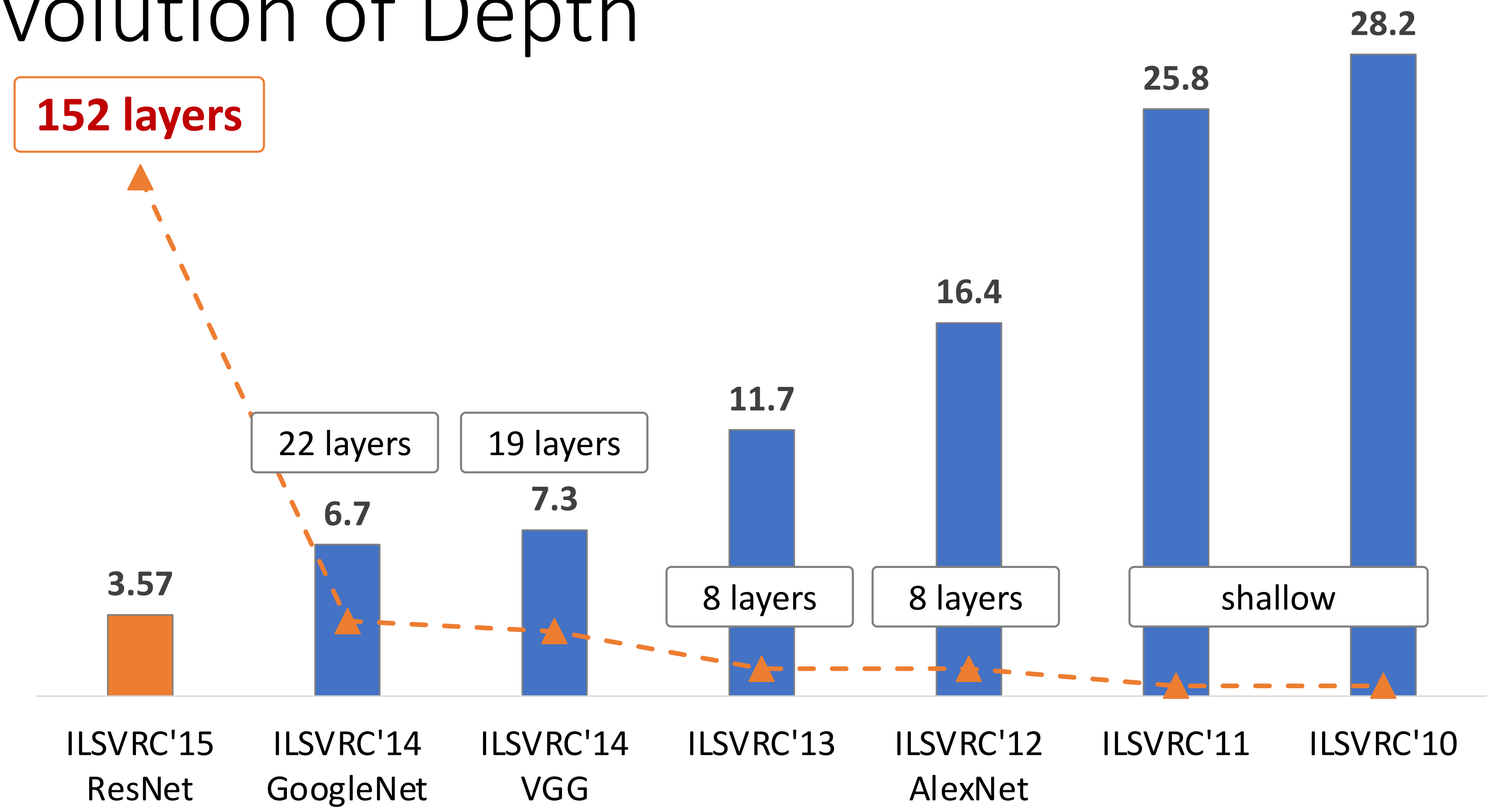
ResNets @ ILSVRC & COCO 2015 Competitions

- **1st places in all five main tracks**
  - ImageNet Classification: "*Ultra-deep*" 152-layer nets
  - ImageNet Detection: 16% better than 2nd
  - ImageNet Localization: 27% better than 2nd
  - COCO Detection: 11% better than 2nd
  - COCO Segmentation: 12% better than 2nd

*improvements are relative numbers

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

Revolution of Depth

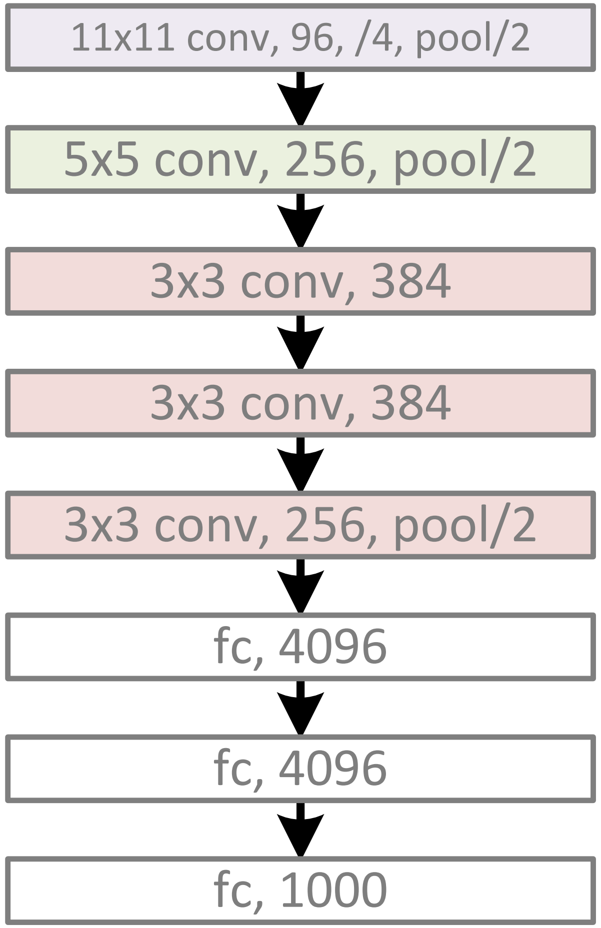


ImageNet Classification top-5 error (%)

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

Revolution of Depth

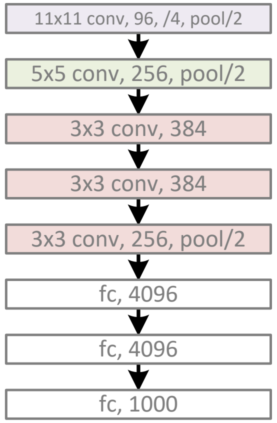
AlexNet, 8 layers
(ILSVRC 2012)



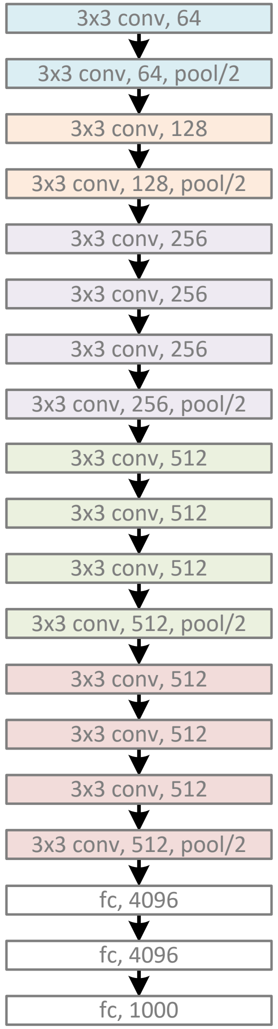
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

Revolution of Depth

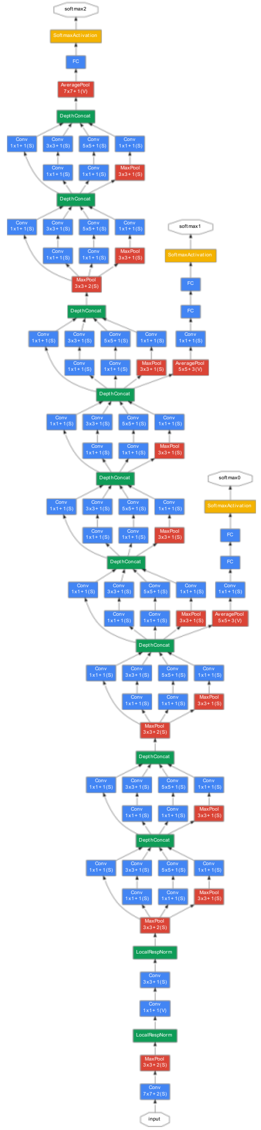
AlexNet, 8 layers
(ILSVRC 2012)



VGG, 19 layers
(ILSVRC 2014)



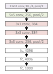
GoogleNet, 22 layers
(ILSVRC 2014)



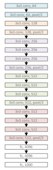
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

Revolution of Depth

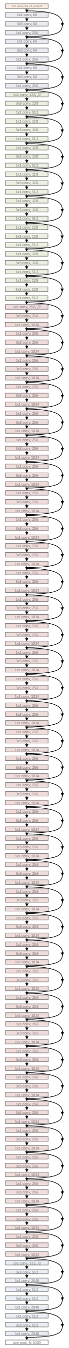
AlexNet, 8 layers
(ILSVRC 2012)



VGG, 19 layers
(ILSVRC 2014)



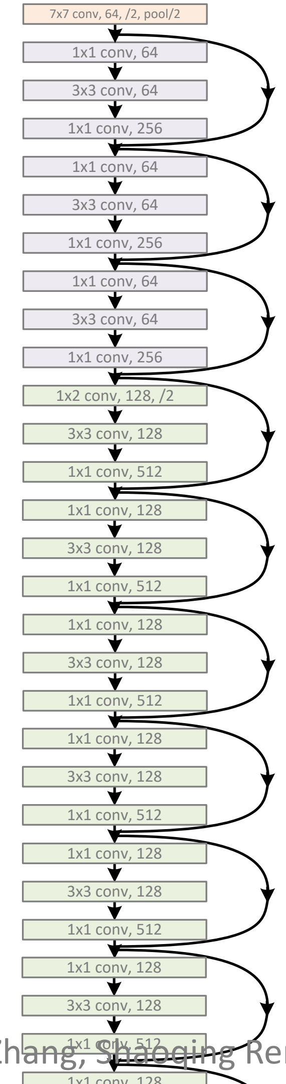
ResNet, 152 layers
(ILSVRC 2015)



Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

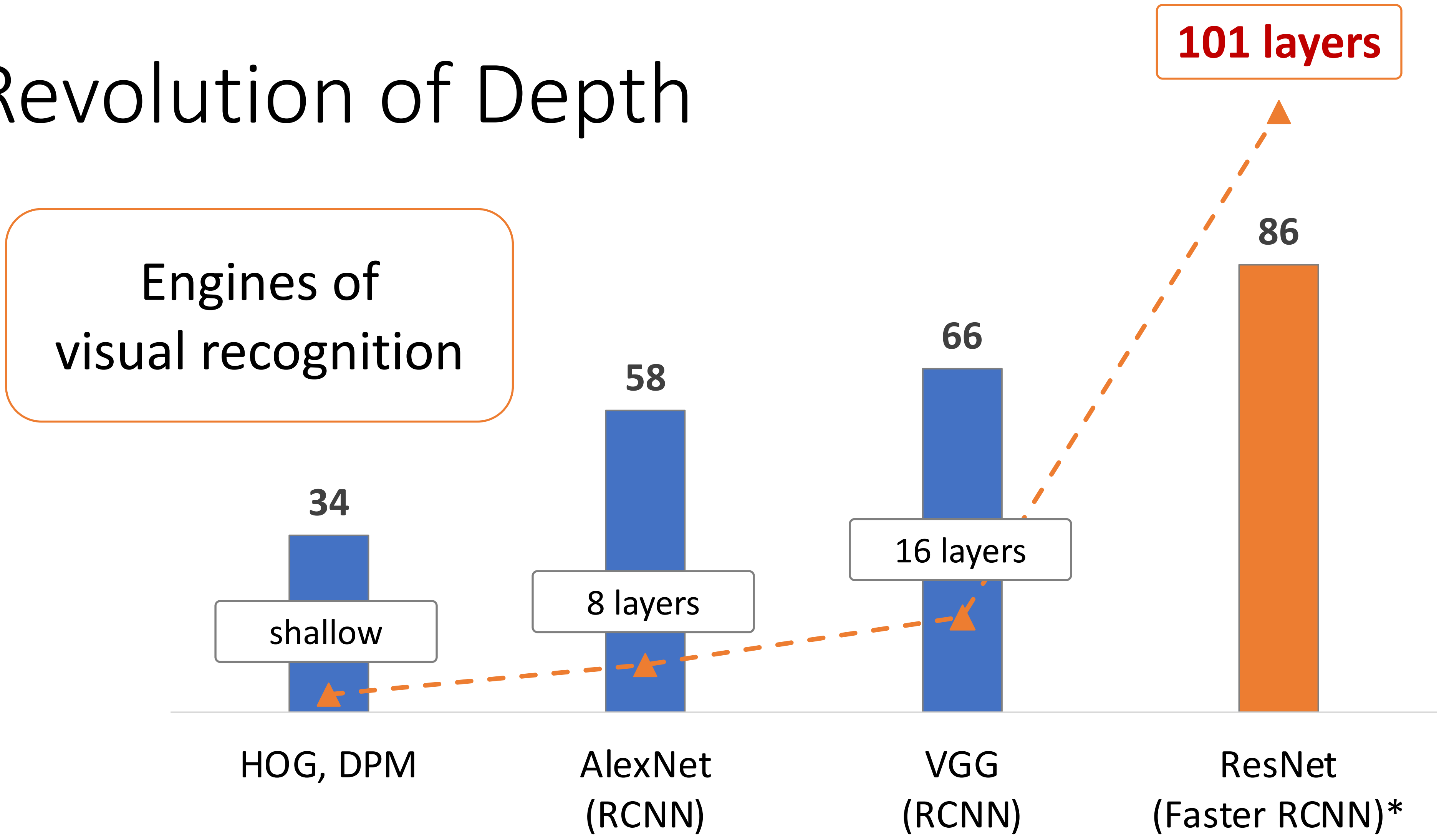
Revolution of Depth

ResNet, 152 layers



Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

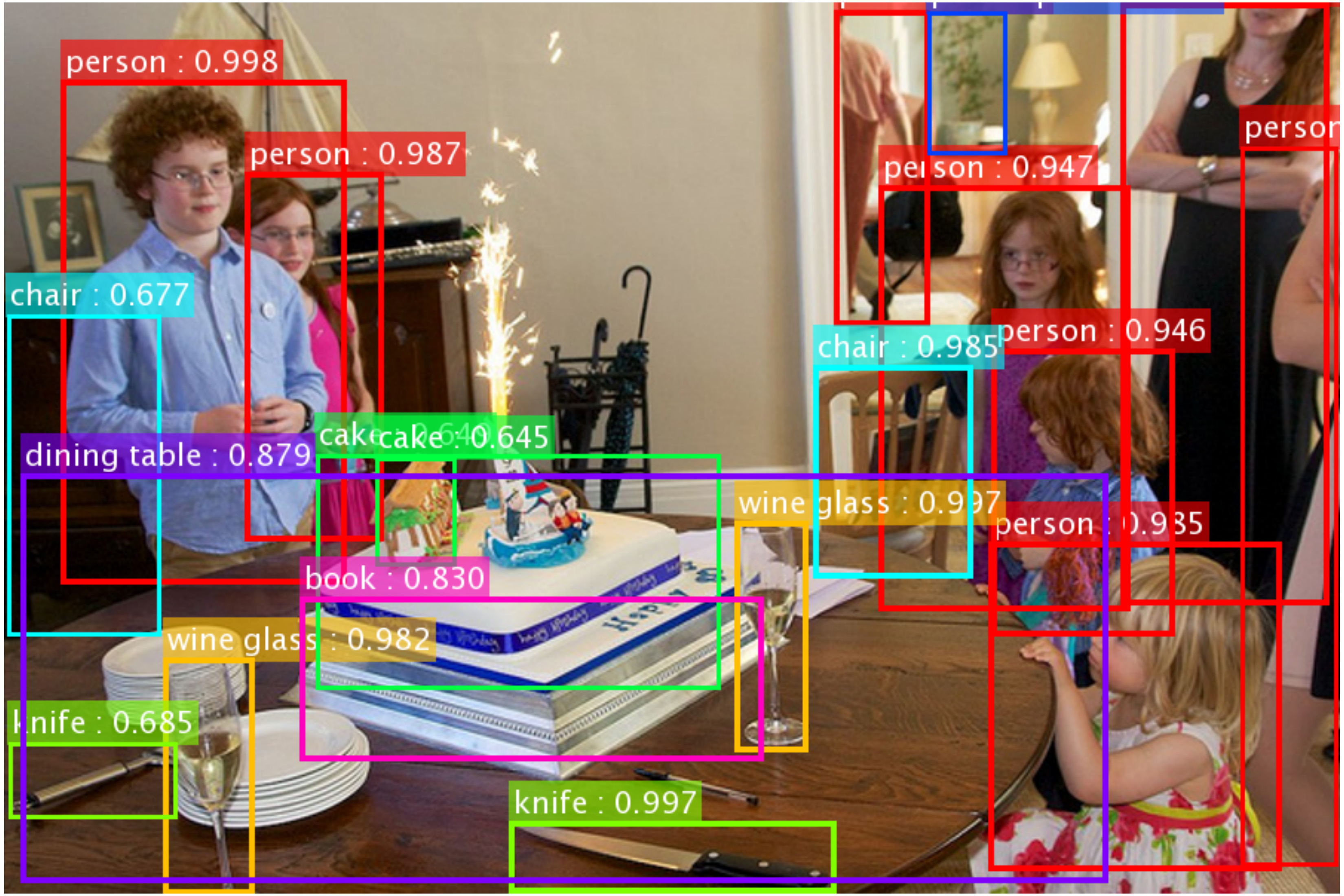
Revolution of Depth



PASCAL VOC 2007 **Object Detection** mAP (%)

*w/ other improvements & more data

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.



ResNet’s object detection result on COCO

*the original image is from the COCO dataset

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

Very simple, easy to follow

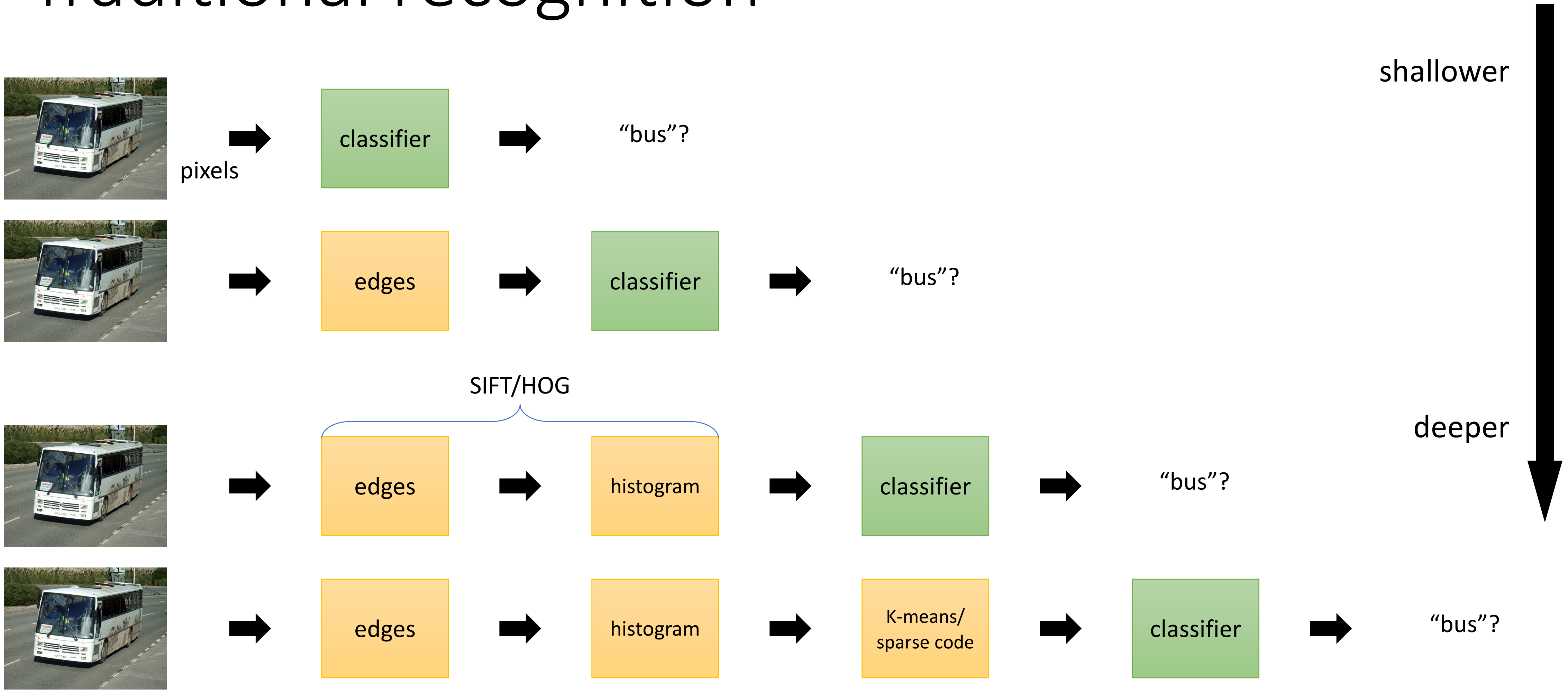
- Many third-party implementations (list in https://github.com/KaimingHe/deep-residual-networks)
  - Facebook AI Research's Torch ResNet:
  - Torch, CIFAR-10, with ResNet-20 to ResNet-110, training code, and curves: code
  - Lasagne, CIFAR-10, with ResNet-32 and ResNet-56 and training code: code
  - Neon, CIFAR-10, with pre-trained ResNet-32 to ResNet-110 models, training code, and curves: code
  - Torch, MNIST, 100 layers: blog, code
  - A winning entry in Kaggle's right whale recognition challenge: blog, code
  - Neon, Place2 (mini), 40 layers: blog, code
  - …
- Easily reproduced results (e.g. Torch ResNet: https://github.com/facebook/fb.resnet.torch)
- A series of extensions and follow-ups
  - > 200 citations in 6 months after posted on arXiv (Dec. 2015)

Background

From shallow to deep

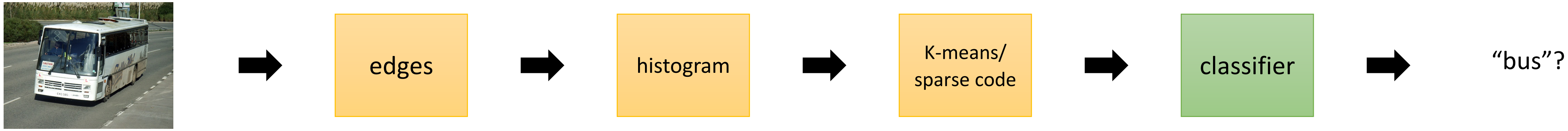
Traditional recognition

But what's next?

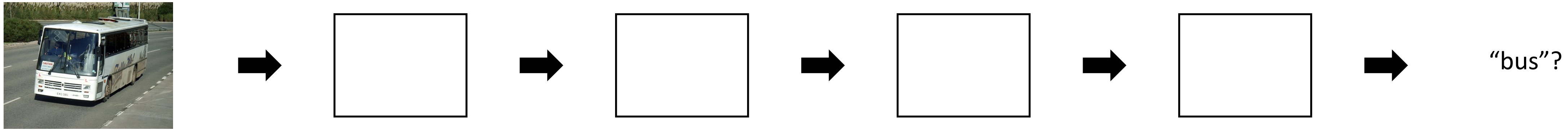


Deep Learning

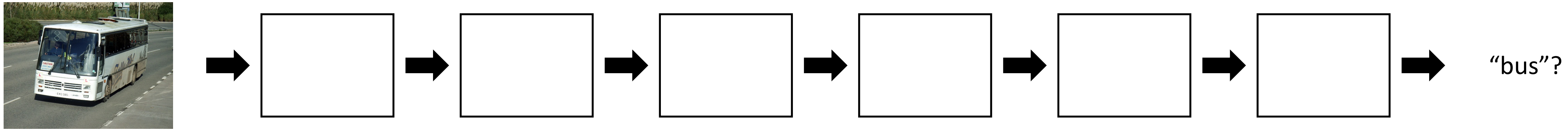
Specialized components, domain knowledge required



Generic components (“layers”), less domain knowledge

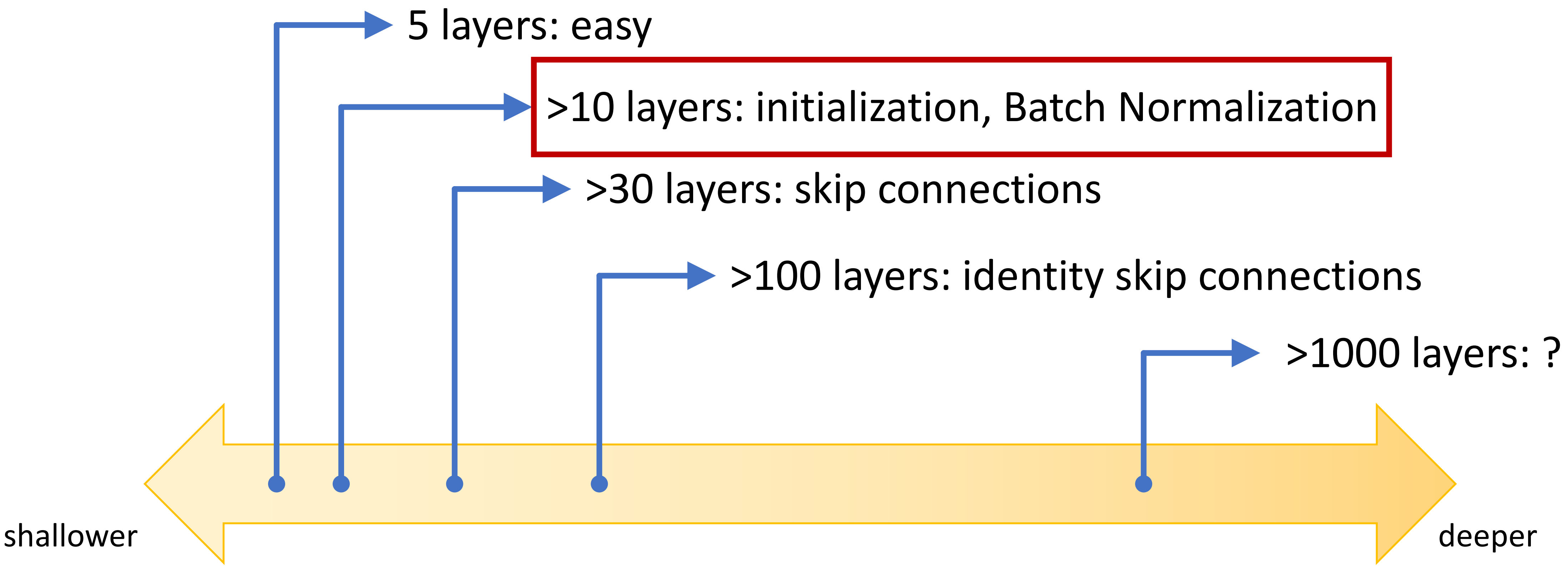


Repeat elementary layers => Going deeper

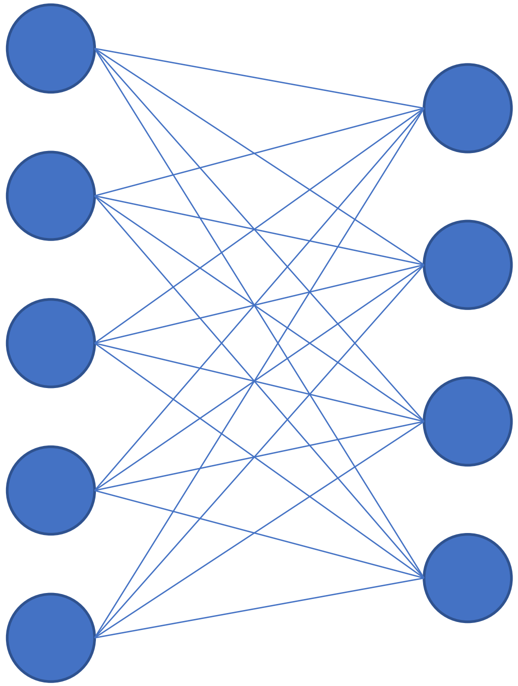


- End-to-end learning
- Richer solution space

Spectrum of Depth



Initialization



If:

- Linear activation
- x, y, w: independent

Then:

1-layer:

$$Var[y] = (n^{in} Var[w]) Var[x]$$

Multi-layer:

$$Var[y] = \left(\prod_d n_d^{in} Var[w_d]\right) Var[x]$$

LeCun et al 1998 “Efficient Backprop”

Glorot & Bengio 2010 “Understanding the difficulty of training deep feedforward neural networks”

Initialization

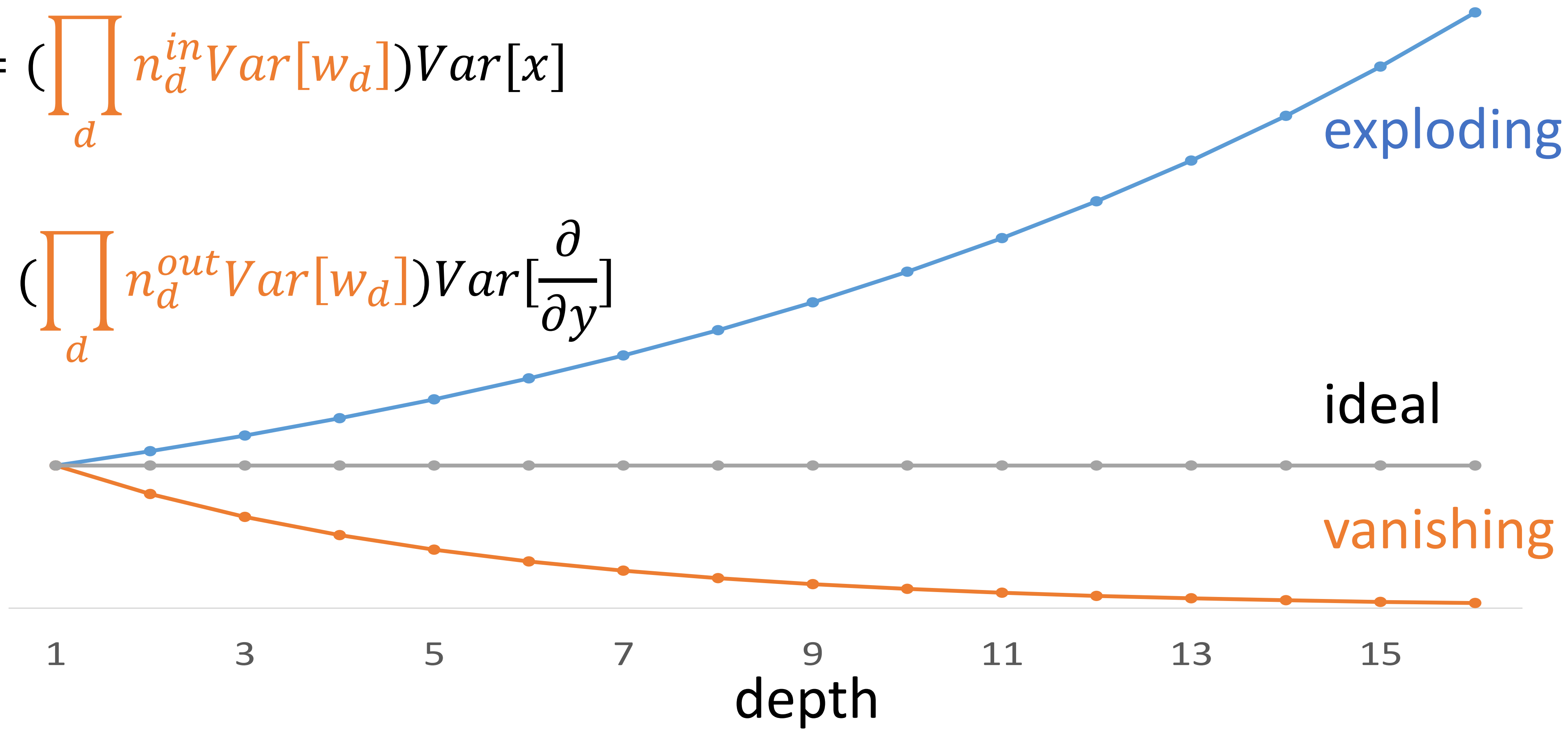
Both forward (response) and backward (gradient) signal can vanish/explode

Forward:

$$Var[y] = (\prod_d n_d^{in} Var[w_d]) Var[x]$$

Backward:

$$Var\left[\frac{\partial}{\partial x}\right] = (\prod_d n_d^{out} Var[w_d]) Var[\frac{\partial}{\partial y}]$$



LeCun et al 1998 “Efficient Backprop”

Glorot & Bengio 2010 “Understanding the difficulty of training deep feedforward neural networks”

Initialization

- Initialization under **linear** assumption

$$\prod_d n_d^{in} Var[w_d] = const_{\text{fw}} \text{ (healthy forward)}$$

and

$$\prod_d n_d^{out} Var[w_d] = const_{\text{bw}} \text{(healthy backward)}$$

➡ $$\boxed{\begin{array}{c} n_d^{in} Var[w_d] = 1 \\ \text{or*} \\ n_d^{out} Var[w_d] = 1 \end{array}}$$

*: $n_d^{out} = n_{d+1}^{in}$, so $\frac{const_{\text{bw}}}{const_{\text{fw}}} = \frac{n_{\text{last}}^{out}}{n_{\text{first}}^{in}} < \infty$.

It is sufficient to use either form.

"*Xavier*" init in Caffe

LeCun et al 1998 "Efficient Backprop"

Glorot & Bengio 2010 "Understanding the difficulty of training deep feedforward neural networks"

Initialization

- Initialization under **ReLU**

$$\prod_d \frac{1}{2} n_d^{in} Var[w_d] = const_{\text{fw}} \text{ (healthy forward)}$$

and

$$\prod_d \frac{1}{2} n_d^{out} Var[w_d] = const_{\text{bw}} \text{(healthy backward)}$$

➡ $$\boxed{\begin{array}{c} \frac{1}{2} n_d^{in} Var[w_d] = 1 \\ \text{or} \\ \frac{1}{2} n_d^{out} Var[w_d] = 1 \end{array}}$$

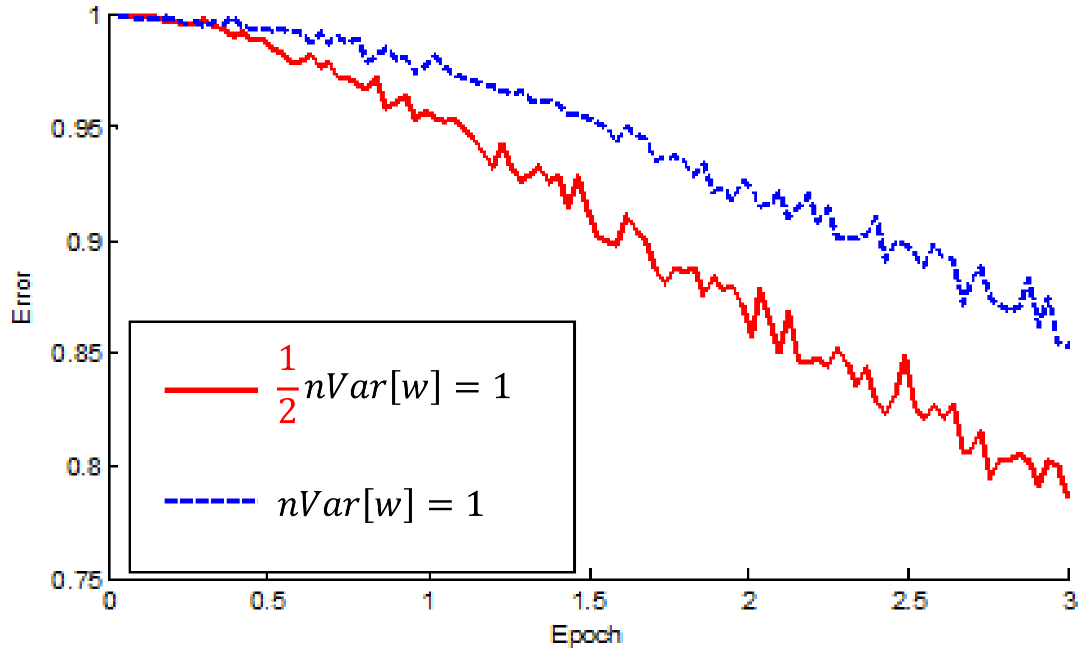
"*MSRA*" init in Caffe

With D layers, a factor of 2 per layer has exponential impact of 2^D

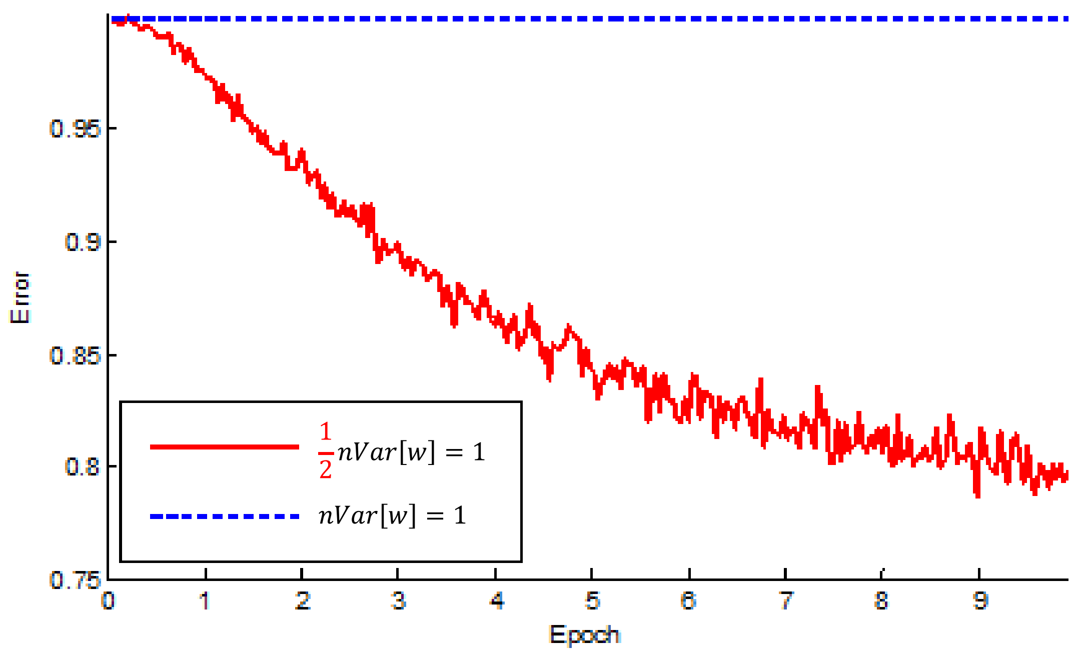
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". ICCV 2015.

Initialization

22-layer ReLU net:
good init converges faster



30-layer ReLU net:
good init is able to converge



*Figures show the beginning of training

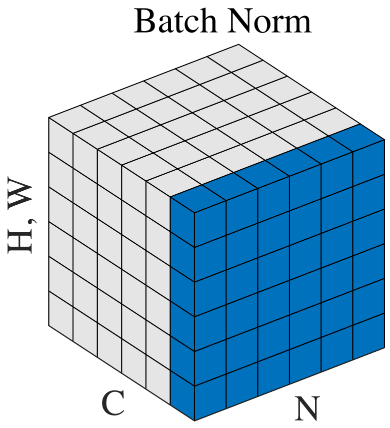
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification”. ICCV 2015.

Batch Normalization (BN)

- Normalizing input (LeCun et al 1998 “Efficient Backprop”)
- BN: normalizing each layer, for each mini-batch
- Greatly accelerate training
- Less sensitive to initialization
- Improve regularization

S. Ioffe & C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. ICML 2015

Batch Normalization (BN)



$$\rightarrow \boxed{\text{layer}} \rightarrow x \rightarrow \quad \hat{x} = \frac{x - \mu}{\sigma} \quad \rightarrow \quad y = \gamma \hat{x} + \beta$$

- μ: mean of x in mini-batch
- σ: std of x in mini-batch
- γ: scale
- β: shift

- μ, σ: functions of x, analogous to responses
- γ, β: parameters to be learned, analogous to weights

S. Ioffe & C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. ICML 2015

Deep Residual Networks

From 10 layers to 100 layers

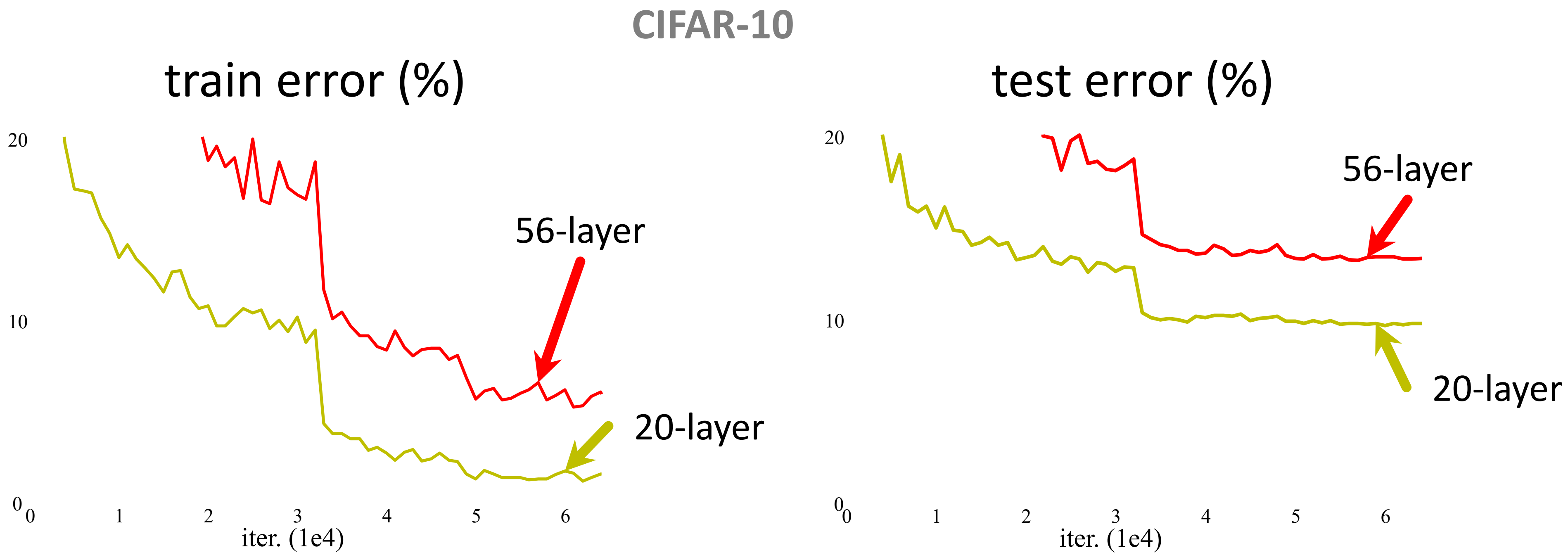
Going Deeper

- Initialization algorithms ✓
- Batch Normalization ✓

- **Is learning better networks as simple as stacking more layers?**

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

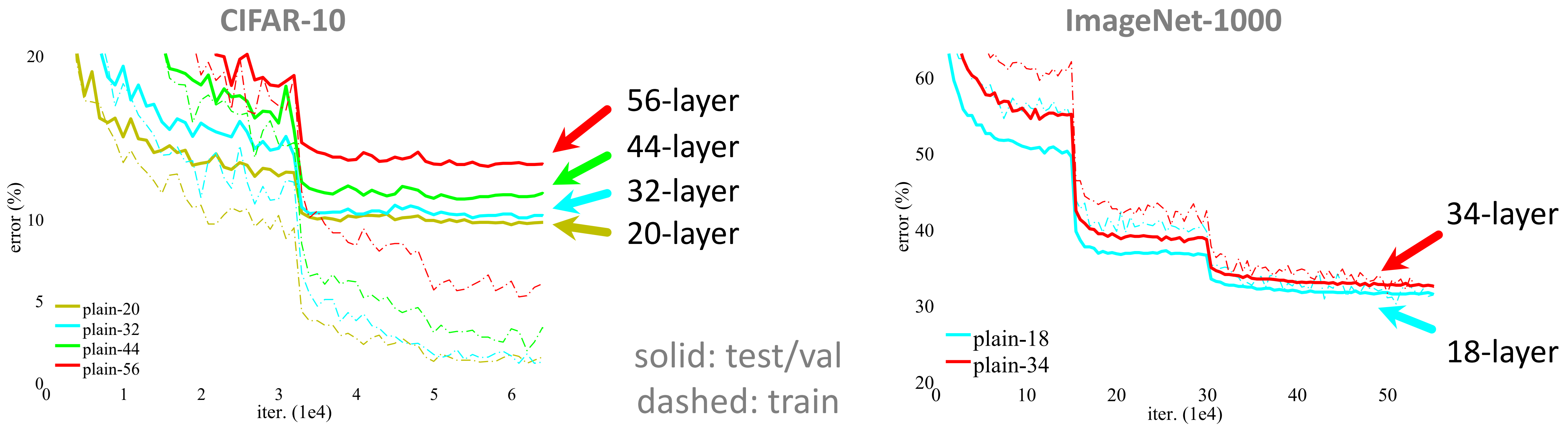
Simply stacking layers?



- *Plain* nets: stacking 3x3 conv layers…
- 56-layer net has **higher training error** and test error than 20-layer net

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

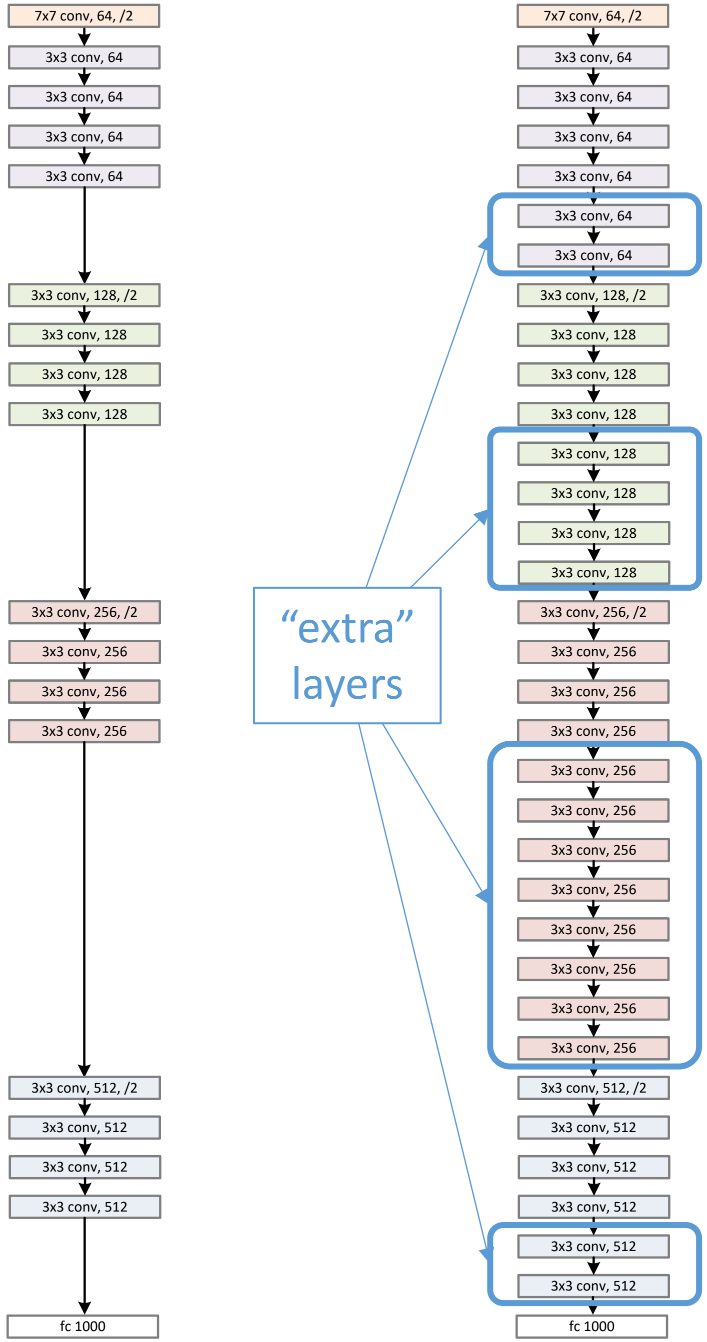
Simply stacking layers?



- "Overly deep" plain nets have **higher training error**
- A general phenomenon, observed in many datasets

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

a shallower model (18 layers)



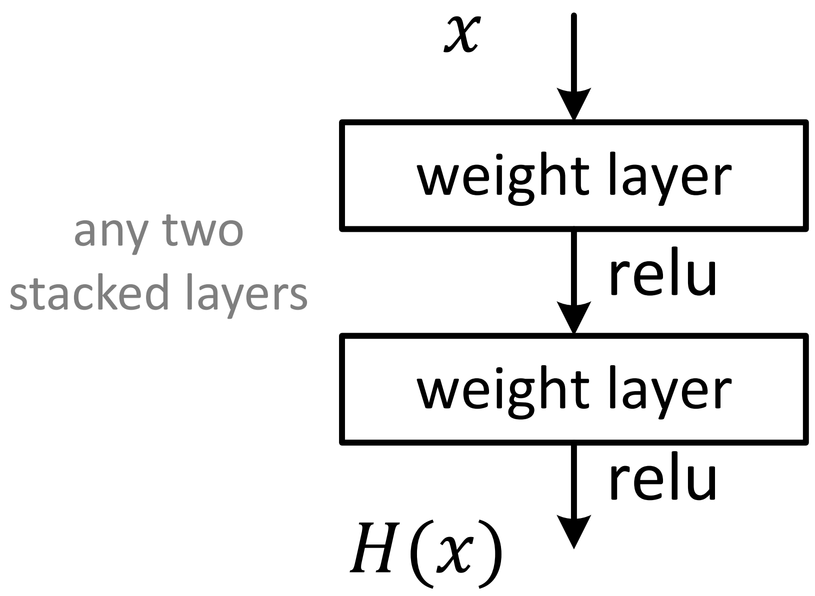
a deeper counterpart (34 layers)

- Richer solution space
- A deeper model should not have **higher training error**
- A solution *by construction*:
  - original layers: copied from a learned shallower model
  - extra layers: set as identity
  - at least the same training error
- Optimization difficulties: solvers cannot find the solution when going deeper…

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

Deep Residual Learning

- Plain net

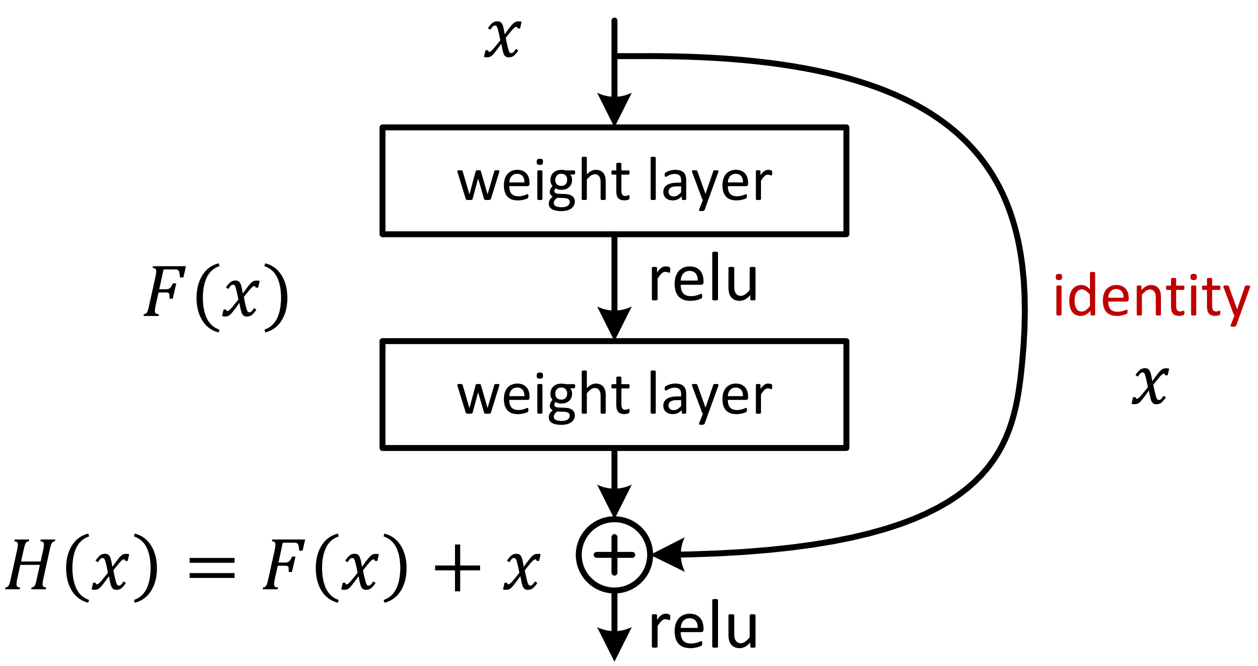


$H(x)$ is any desired mapping,

hope the 2 weight layers fit $H(x)$

Deep Residual Learning

- Residual net



$H(x)$ is any desired mapping,

~~hope the 2 weight layers fit $H(x)$~~

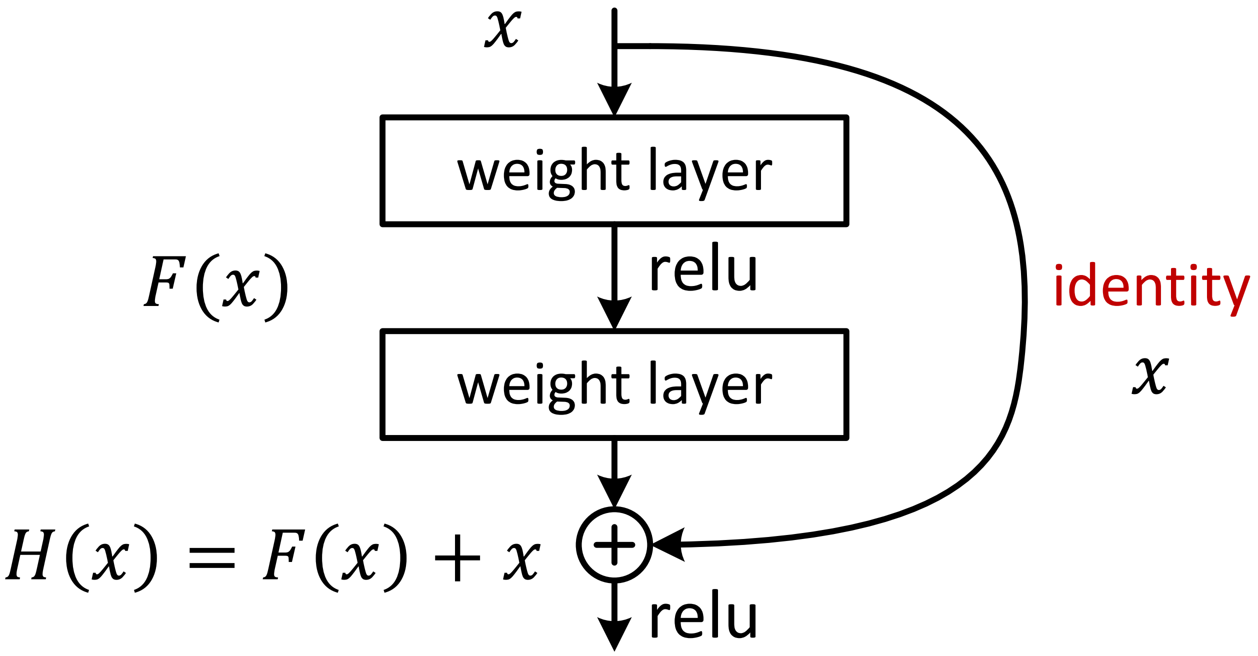
hope the 2 weight layers fit $F(x)$

let $H(x) = F(x) + x$

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

Deep Residual Learning

- $F(x)$ is a residual mapping w.r.t. identity



- If identity were optimal, easy to set weights as 0

- If optimal mapping is closer to identity, easier to find small fluctuations

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

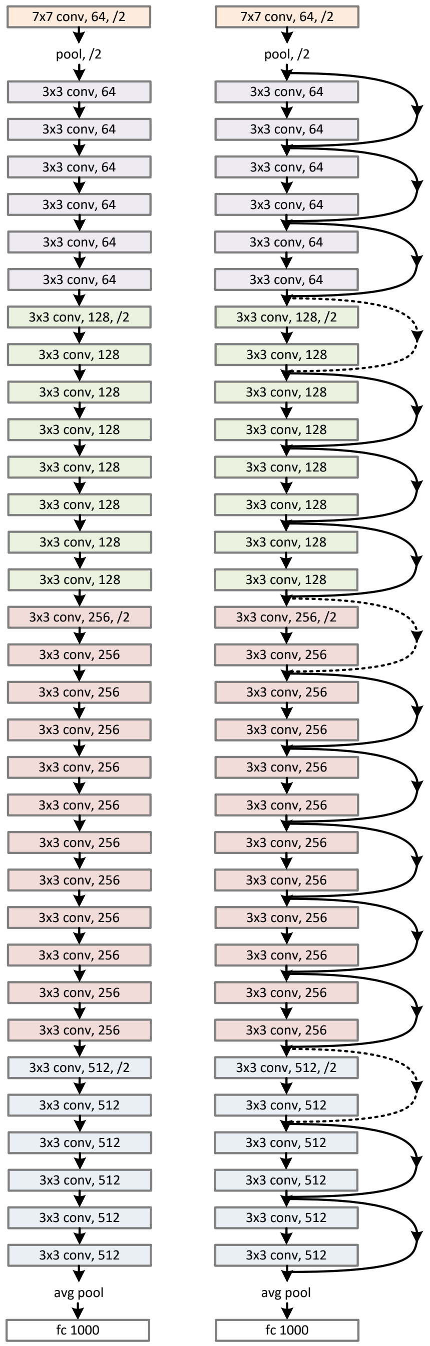
Related Works – Residual Representations

- VLAD & Fisher Vector [Jegou et al 2010], [Perronnin et al 2007]
  - Encoding residual vectors; powerful shallower representations.
- Product Quantization (IVF-ADC) [Jegou et al 2011]
  - Quantizing residual vectors; efficient nearest-neighbor search.
- MultiGrid & Hierarchical Precondition [Briggs, et al 2000], [Szeliski 1990, 2006]
  - Solving residual sub-problems; efficient PDE solvers.

Network "Design"

- Keep it simple
- Our basic design (VGG-style)
  - all 3x3 conv (almost)
  - spatial size /2 => # filters x2 (~same complexity per layer)
  - Simple design; just deep!
- Other remarks:
  - no hidden fc
  - no dropout

plain net



ResNet

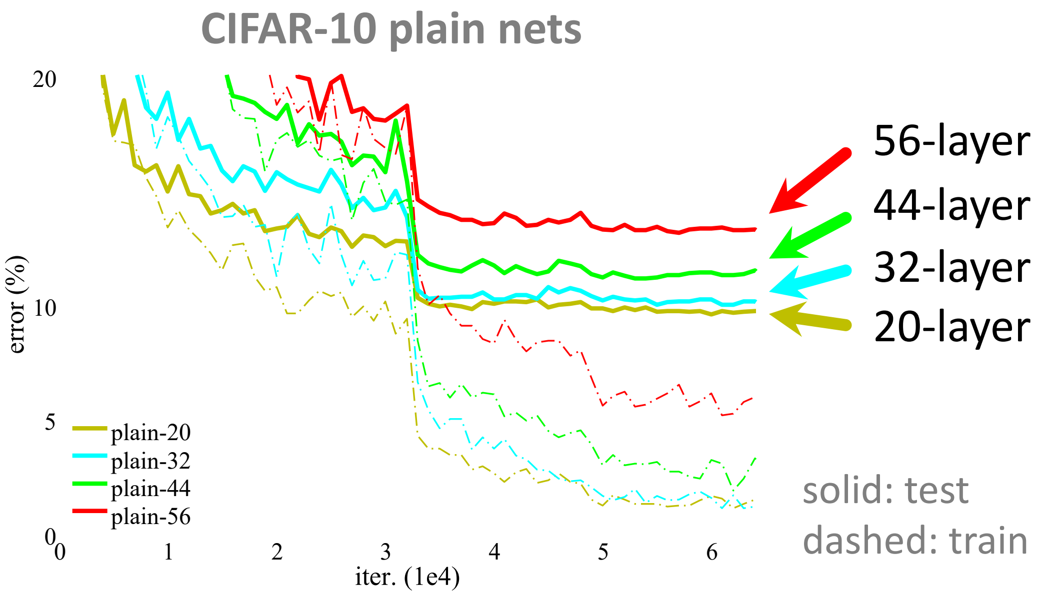
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

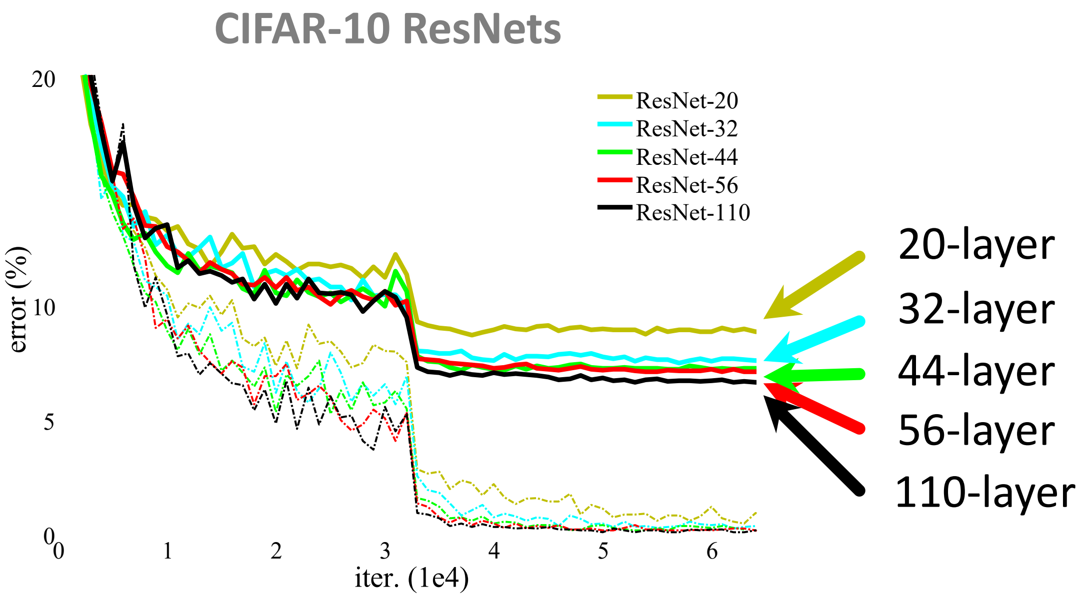
Training

- All plain/residual nets are trained from scratch
- All plain/residual nets use Batch Normalization
- Standard hyper-parameters & augmentation

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

CIFAR-10 experiments

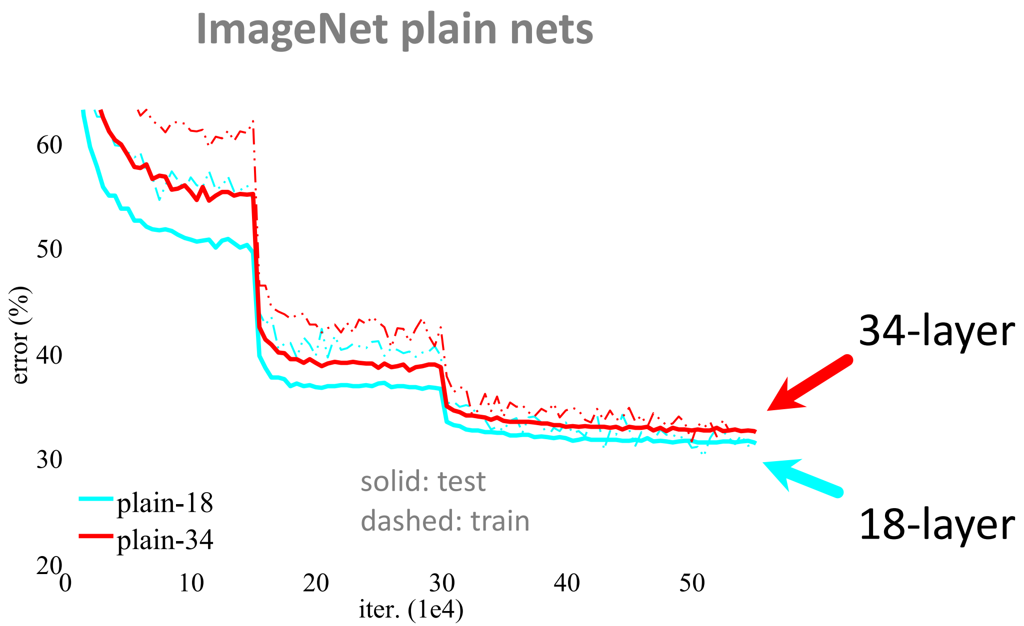


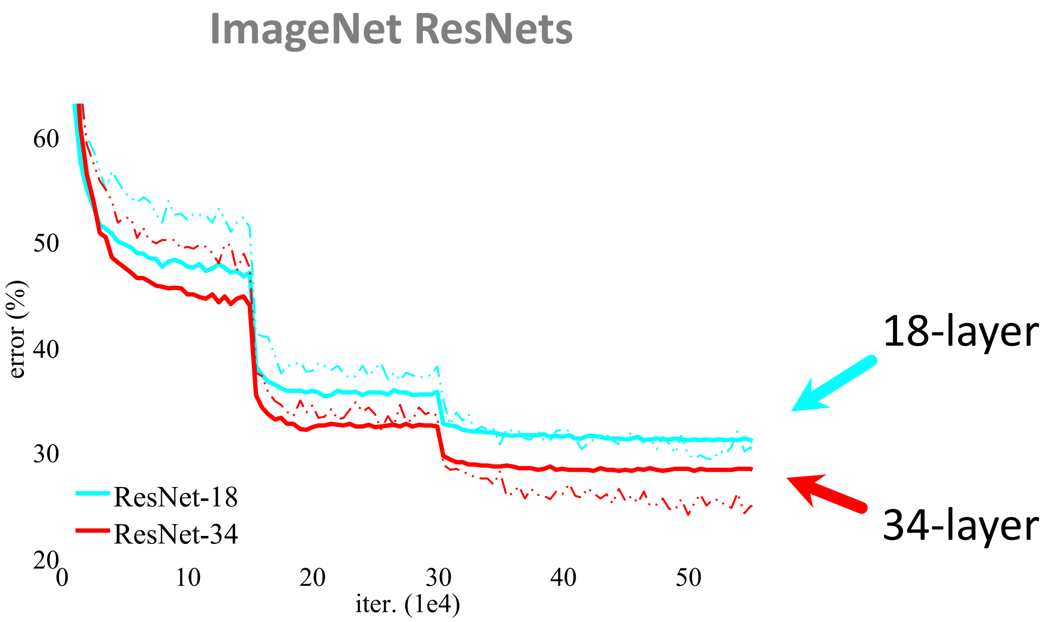


- Deep ResNets can be trained without difficulties
- Deeper ResNets have **lower training error**, and also lower test error

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

ImageNet experiments



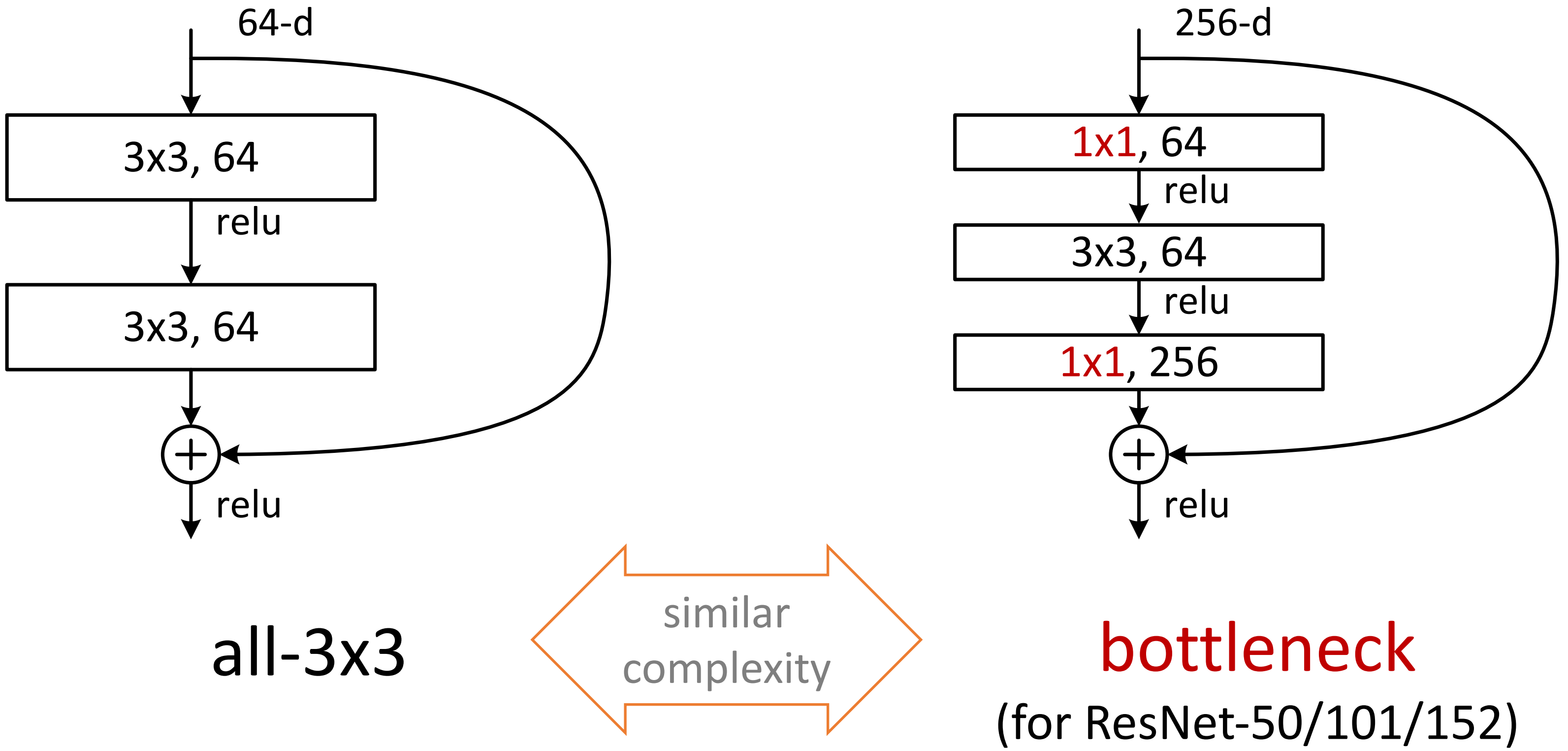


- Deep ResNets can be trained without difficulties
- Deeper ResNets have **lower training error**, and also lower test error

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

ImageNet experiments

- A practical design of going deeper

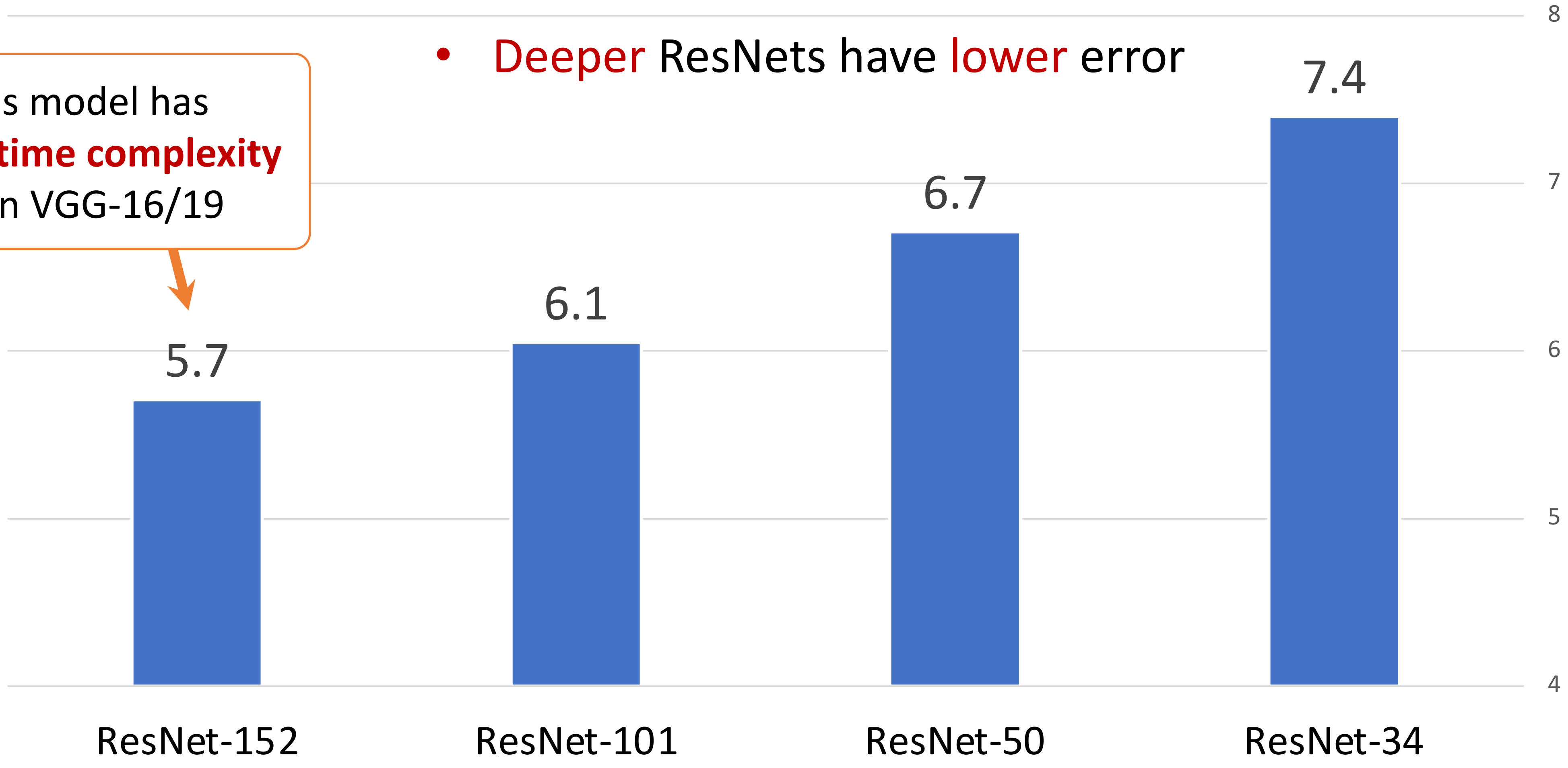


Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

ImageNet experiments

- **Deeper** ResNets have **lower** error

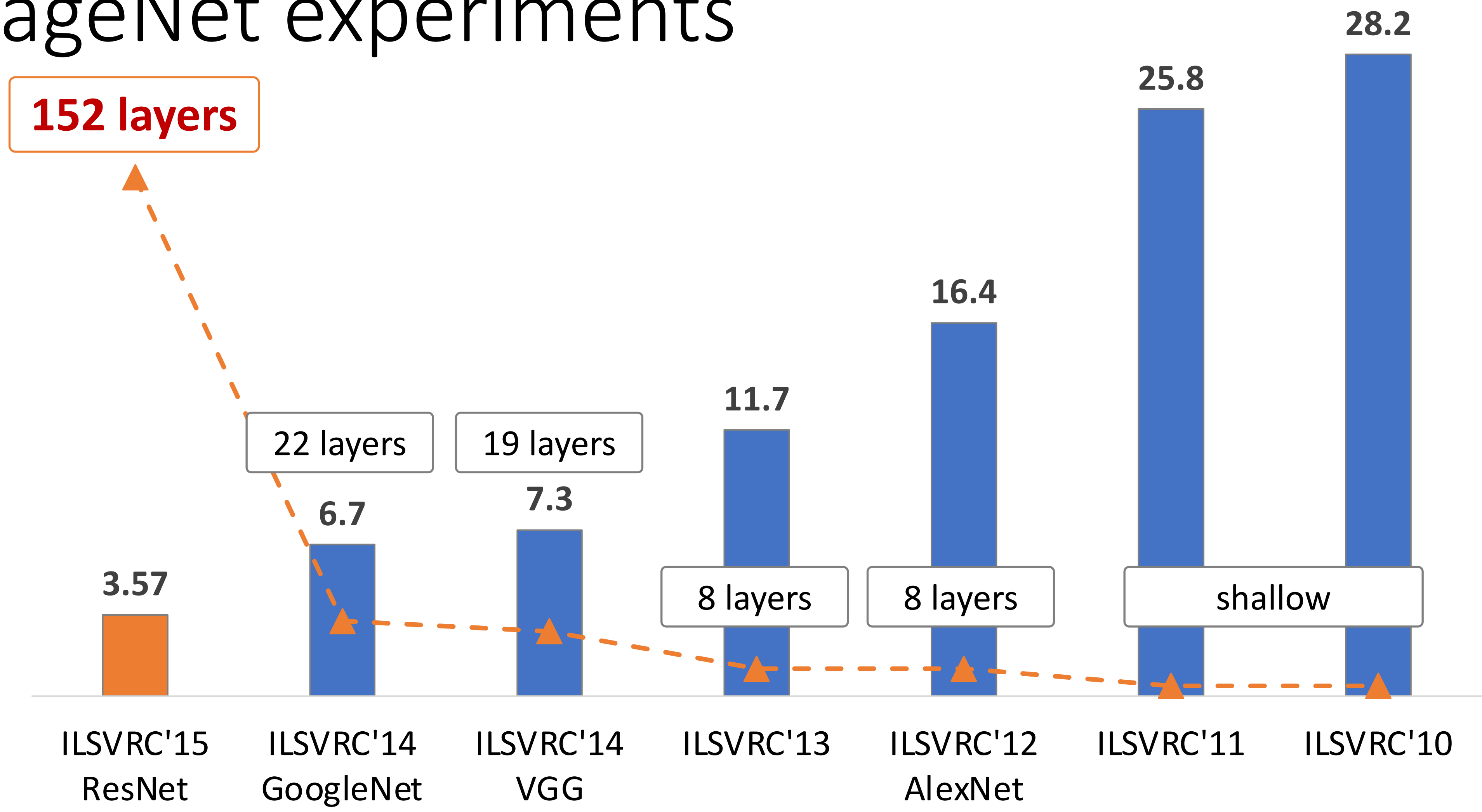
this model has
lower time complexity
than VGG-16/19



10-crop testing, top-5 val error (%)

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

ImageNet experiments



ImageNet Classification top-5 error (%)

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

Discussions

Representation, Optimization, Generalization

Issues on learning deep models

- **Representation** ability
  - Ability of model to fit training data, if optimum could be found
  - If model A's solution space is a superset of B's, A should be better.
- **Optimization** ability
  - Feasibility of finding an optimum
  - Not all models are equally easy to optimize
- **Generalization** ability
  - Once training data is fit, how good is the test performance

Léon Bottou and Olivier Bousquet: **The Tradeoffs of Large Scale Learning**, *Advances in Neural Information Processing Systems 20 (NIPS 2007)*,

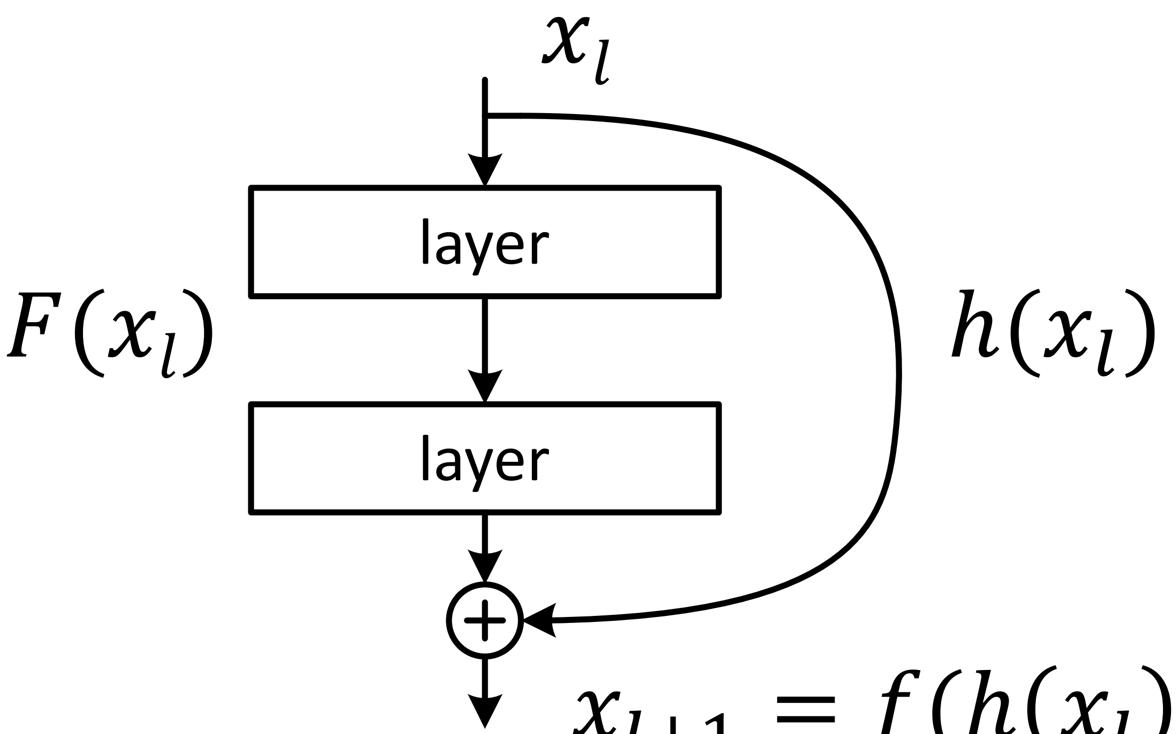
How do ResNets address these issues?

- **Representation** ability
  - No explicit advantage on representation (only re-parameterization), but
  - Allow models to go deeper
- **Optimization** ability
  - Enable very smooth forward/backward prop
  - Greatly ease optimizing deeper models
- **Generalization** ability
  - Not explicitly address generalization, but
  - Deeper+thinner is good generalization

On the Importance of Identity Mapping

From 100 layers to 1000 layers

On identity mappings for **optimization**

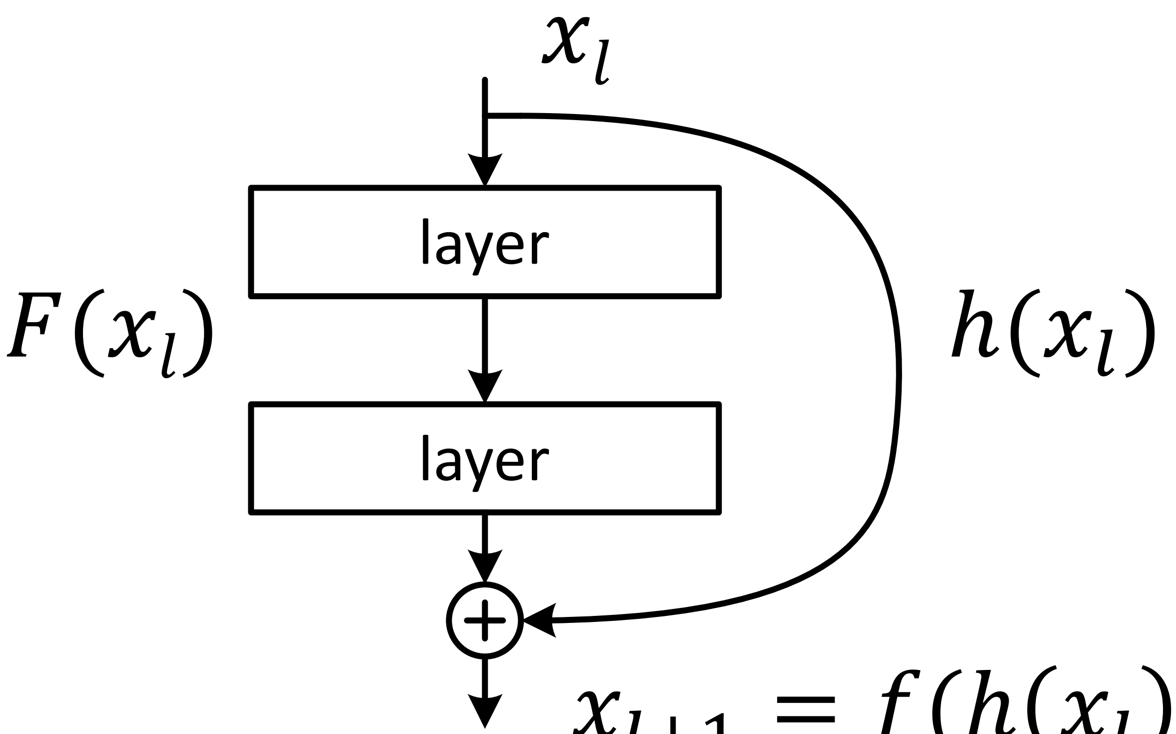


$$x_{l+1} = f(h(x_l) + F(x_l))$$

- shortcut mapping: h = identity
- after-add mapping: f = ReLU

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Identity Mappings in Deep Residual Networks”. arXiv 2016.

On identity mappings for **optimization**

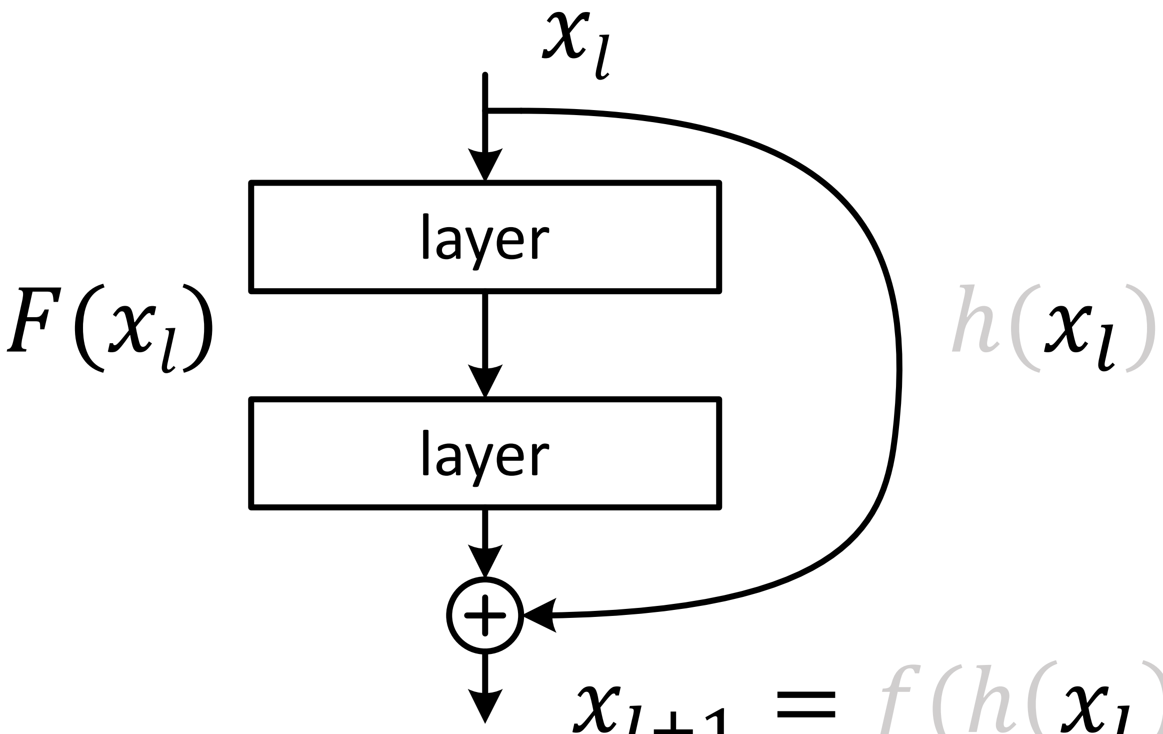


$$x_{l+1} = f(h(x_l) + F(x_l))$$

- shortcut mapping: h = identity
- ~~after-add mapping: f = ReLU~~
- What if f = identity?

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Identity Mappings in Deep Residual Networks". arXiv 2016.

On identity mappings for **optimization**



$$x_{l+1} = f(h(x_l) + F(x_l))$$

- shortcut mapping: h = identity
- ~~after-add mapping: f = ReLU~~
- What if f = identity?

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Identity Mappings in Deep Residual Networks”. arXiv 2016.

Very smooth forward propagation

$$x_{l+1} = x_l + F(x_l)$$

$$\Downarrow$$

$$x_{l+2} = x_{l+1} + F(x_{l+1})$$

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Identity Mappings in Deep Residual Networks”. arXiv 2016.

Very smooth forward propagation

$$x_{l+1} = x_l + F(x_l)$$

$$x_{l+2} = x_{l+1} + F(x_{l+1})$$

$$x_{l+2} = x_l + F(x_l) + F(x_{l+1})$$

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Identity Mappings in Deep Residual Networks". arXiv 2016.

Very smooth forward propagation

$$x_{l+1} = x_l + F(x_l)$$

$$x_{l+2} = x_{l+1} + F(x_{l+1})$$

$$x_{l+2} = x_l + F(x_l) + F(x_{l+1})$$

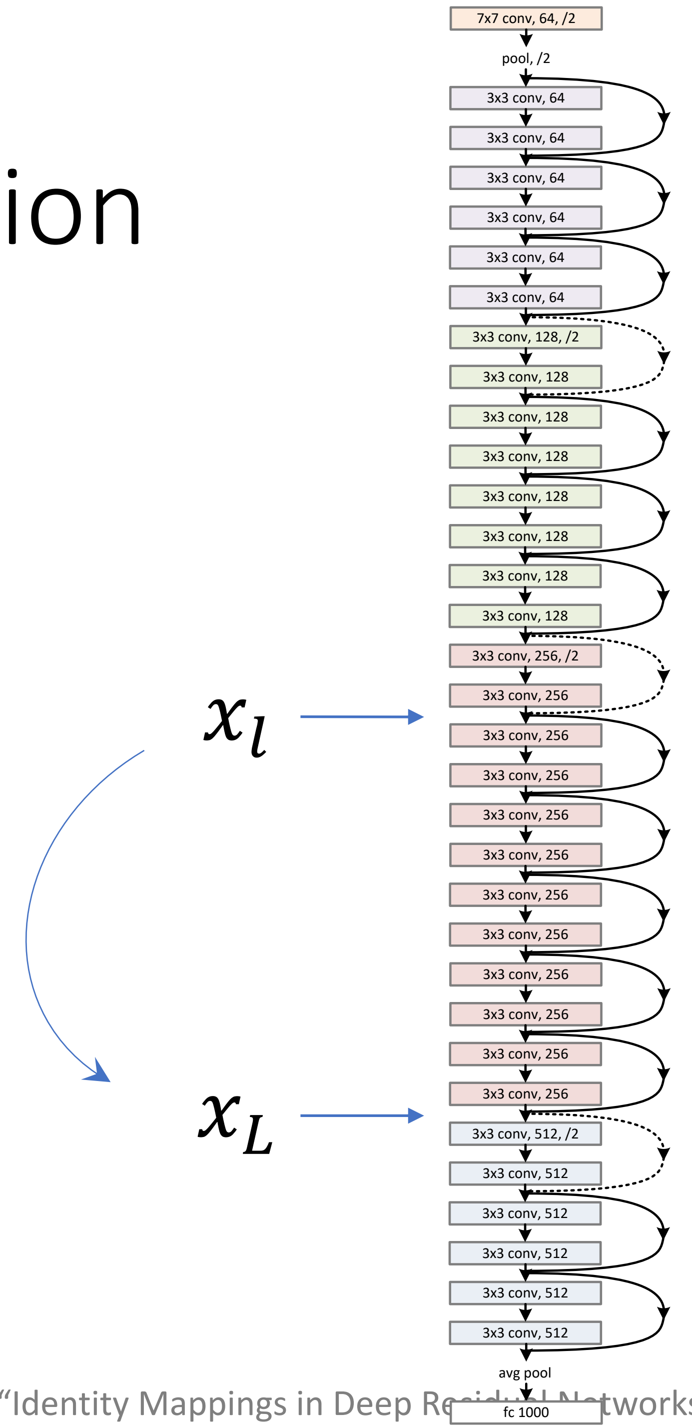
$$x_L = x_l + \sum_{i=l}^{L-1} F(x_i)$$

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Identity Mappings in Deep Residual Networks". arXiv 2016.

Very smooth forward propagation

$$x_L = x_l + \sum_{i=l}^{L-1} F(x_i)$$

- Any x_l is directly forward-prop to any x_L, plus residual.
- Any x_L is an additive outcome.
  - in contrast to multiplicative: $x_L = \prod_{i=l}^{L-1} W_i x_l$



Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Identity Mappings in Deep Residual Networks”. arXiv 2016.

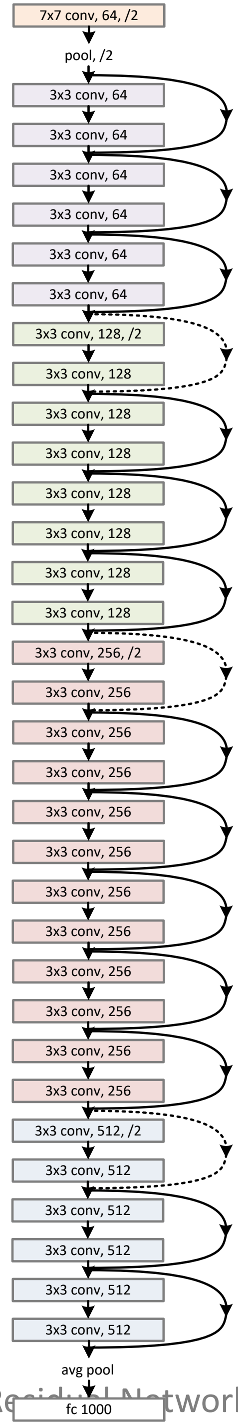
Very smooth backward propagation

$$x_L = x_l + \sum_{i=l}^{L-1} F(x_i)$$

$$\frac{\partial E}{\partial x_l} = \frac{\partial E}{\partial x_L}\frac{\partial x_L}{\partial x_l} = \frac{\partial E}{\partial x_L}\left(1 + \frac{\partial}{\partial x_l}\sum_{i=1}^{L-1} F(x_i)\right)$$

$$\frac{\partial E}{\partial x_l}$$

$$\frac{\partial E}{\partial x_L}$$

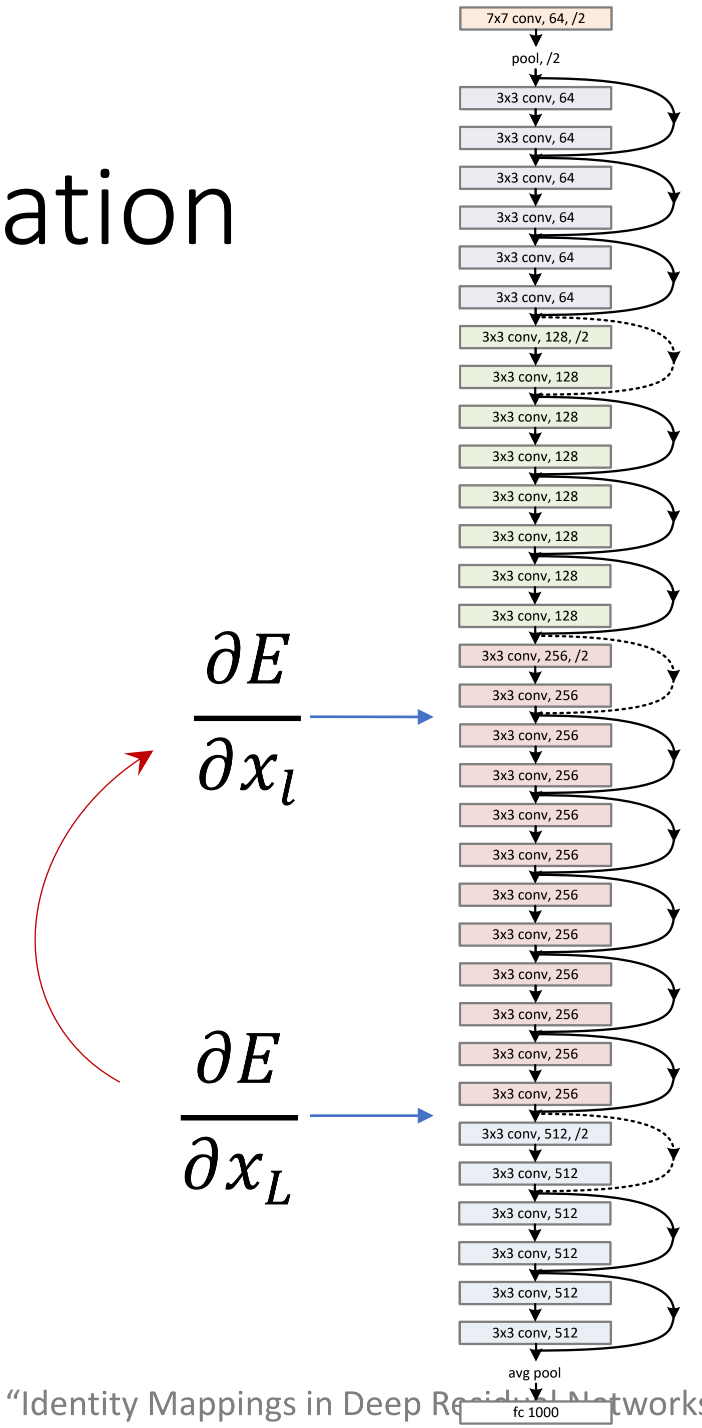


Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Identity Mappings in Deep Residual Networks". arXiv 2016.

Very smooth backward propagation

$$\frac{\partial E}{\partial x_l} = \frac{\partial E}{\partial x_L}\left(1 + \frac{\partial}{\partial x_l}\sum_{i=1}^{L-1} F(x_i)\right)$$

- Any $\frac{\partial E}{\partial x_L}$ is directly back-prop to any $\frac{\partial E}{\partial x_l}$, plus residual.
- Any $\frac{\partial E}{\partial x_l}$ is additive; unlikely to vanish
  - in contrast to multiplicative: $\frac{\partial E}{\partial x_l} = \prod_{i=l}^{L-1} W_i \frac{\partial E}{\partial x_L}$



Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Identity Mappings in Deep Residual Networks". arXiv 2016.

Residual for every layer

forward: $x_L = x_l + \sum_{i=l}^{L-1} F(x_i)$

Enabled by:

- shortcut mapping: h = identity
- after-add mapping: f = identity

backward: $\frac{\partial E}{\partial x_l} = \frac{\partial E}{\partial x_L}\left(1 + \frac{\partial}{\partial x_l}\sum_{i=1}^{L-1} F(x_i)\right)$

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Identity Mappings in Deep Residual Networks". arXiv 2016.

Experiments

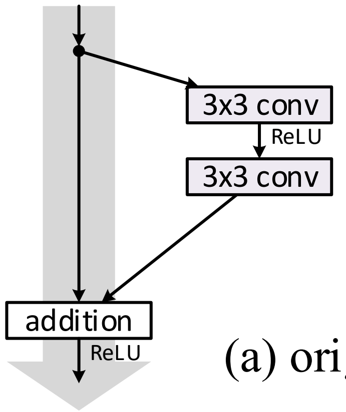
- Set 1: what if shortcut mapping $h \neq$ identity
- Set 2: what if after-add mapping f is identity
- Experiments on ResNets with more than 100 layers
    - deeper models suffer more from optimization difficulty

Experiment Set 1:
what if shortcut mapping $h \neq$ identity?

* ResNet-110 on CIFAR-10

$$h(x) = x$$

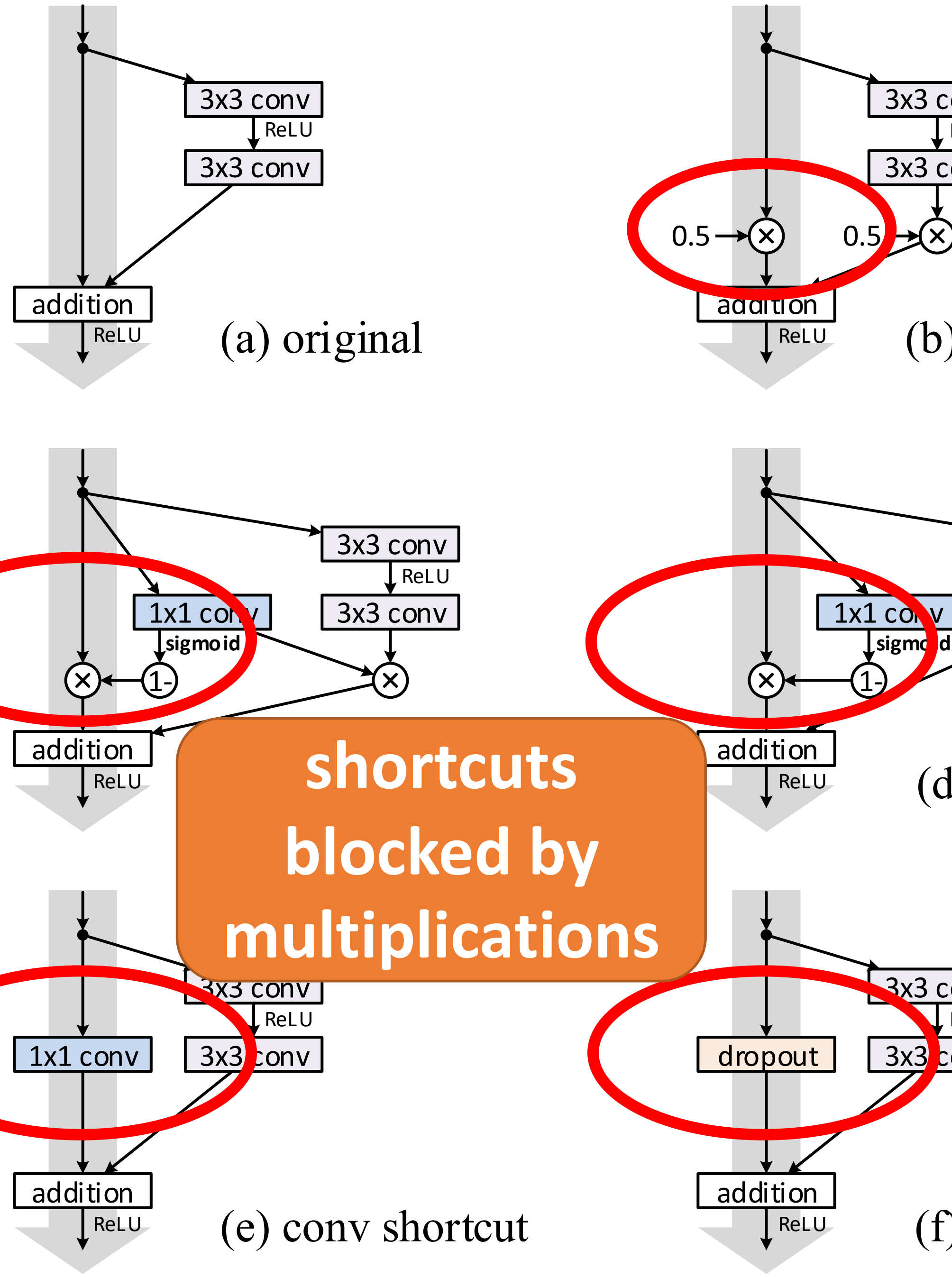
error: 6.6%



(a) original

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Identity Mappings in Deep Residual Networks”. arXiv 2016.

* ResNet-110 on CIFAR-10



(a) original

$h(x) = x$

error: 6.6%

(b) constant scaling

$h(x) = 0.5x$

error: 12.4%

$h(x) = \text{gate} \cdot x$

error: 8.7%

(d) shortcut-only gating

$h(x) = \text{gate} \cdot x$

error: 12.9%

(e) conv shortcut

$h(x) = \text{conv}(x)$

error: 12.2%

(f) dropout shortcut

$h(x) = \text{dropout}(x)$

error: > 20%

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Identity Mappings in Deep Residual Networks". arXiv 2016.

If h is multiplicative, e.g. $h(x) = \lambda x$

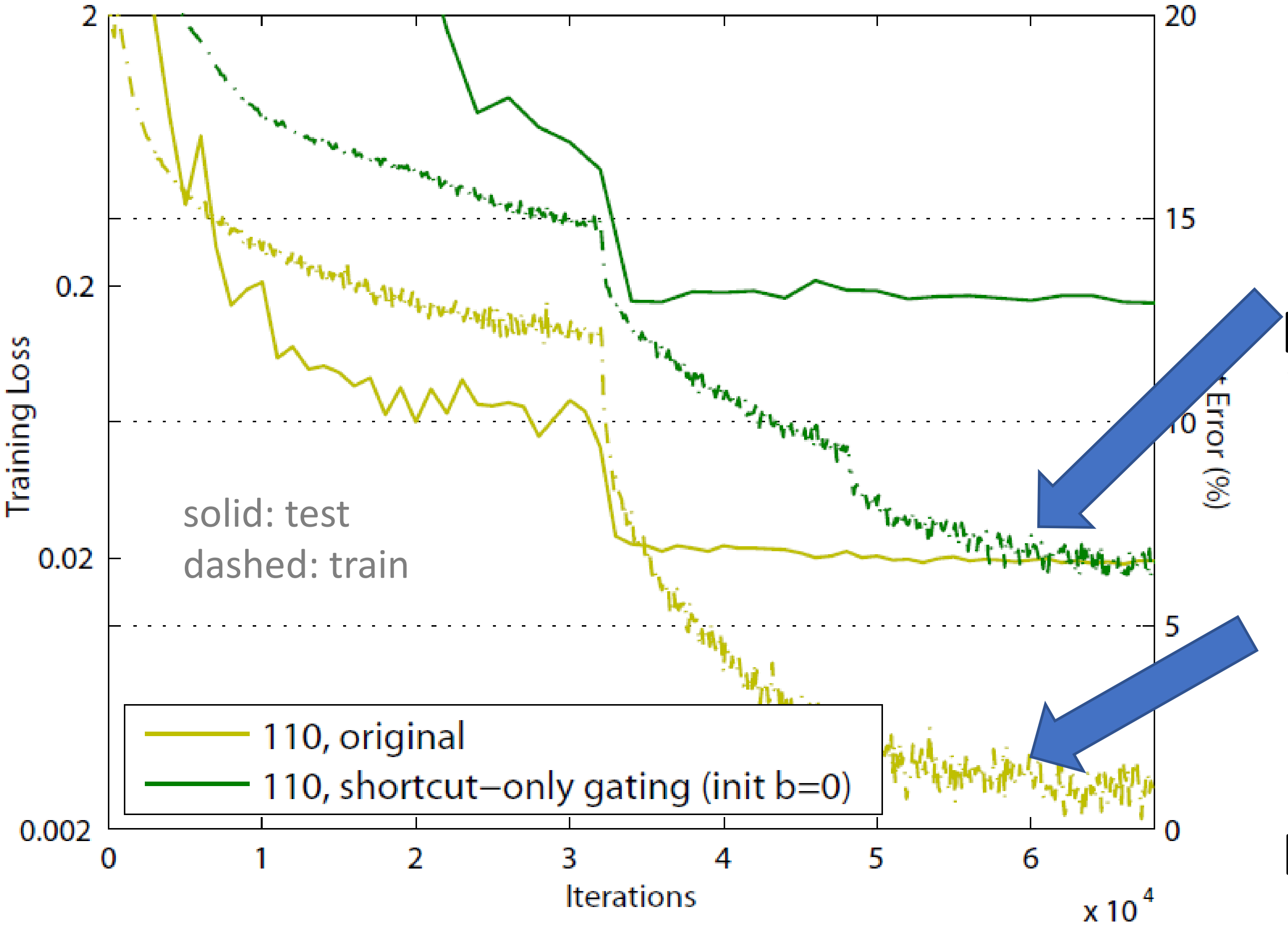
forward: $$x_L = \lambda^{L-l} x_l + \sum_{i=l}^{L-1} \hat{F}(x_i)$$

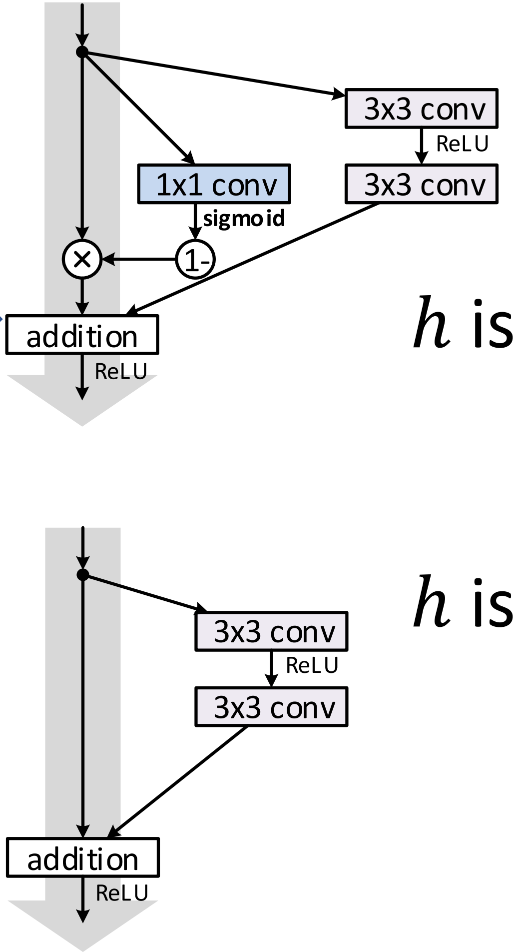
- if h is multiplicative, shortcuts are blocked
- direct propagation is decayed

backward: $$\frac{\partial E}{\partial x_l} = \frac{\partial E}{\partial x_L} \left(\lambda^{L-l} + \frac{\partial}{\partial x_l} \sum_{i=1}^{L-1} \hat{F}(x_i)\right)$$

*assuming f = identity

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Identity Mappings in Deep Residual Networks". arXiv 2016.





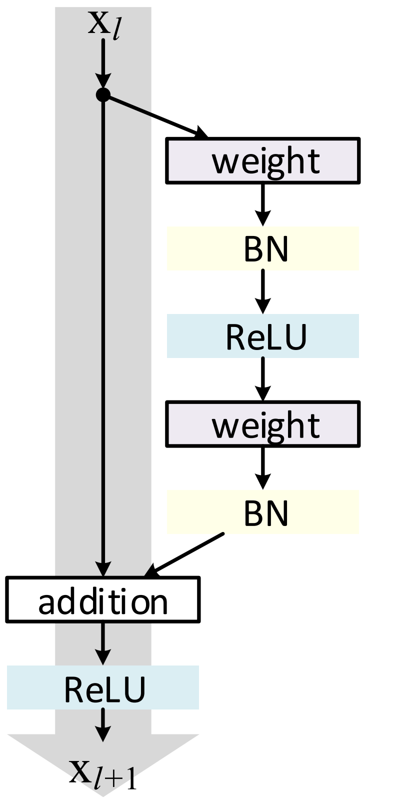
h is gating

h is identity

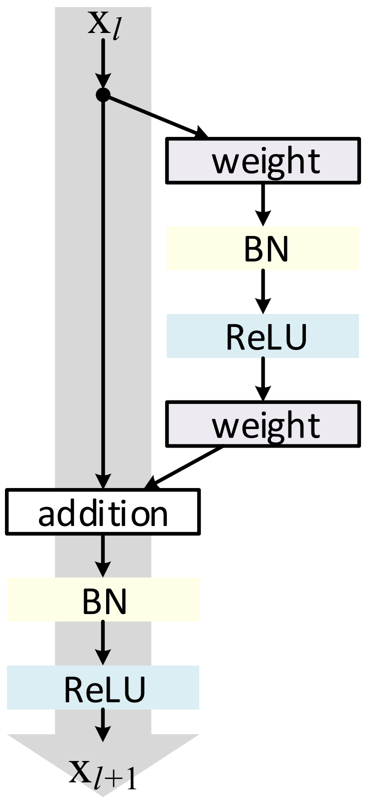
- gating should have better representation ability (identity is a special case), but
- optimization difficulty dominates results

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Identity Mappings in Deep Residual Networks”. arXiv 2016.

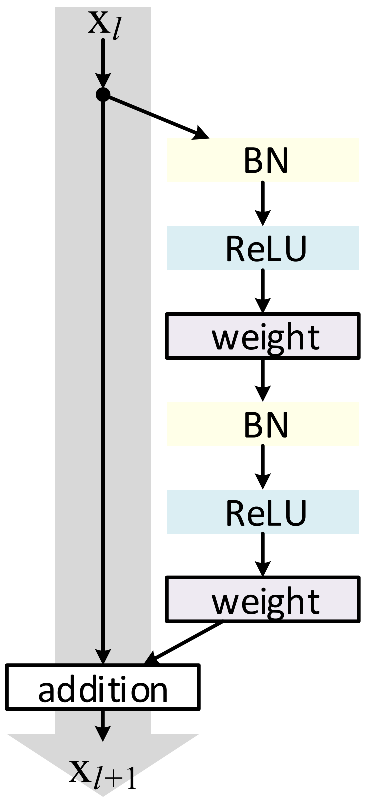
Experiment Set 2:
what if after-add mapping f is identity



f is ReLU
(original ResNet)

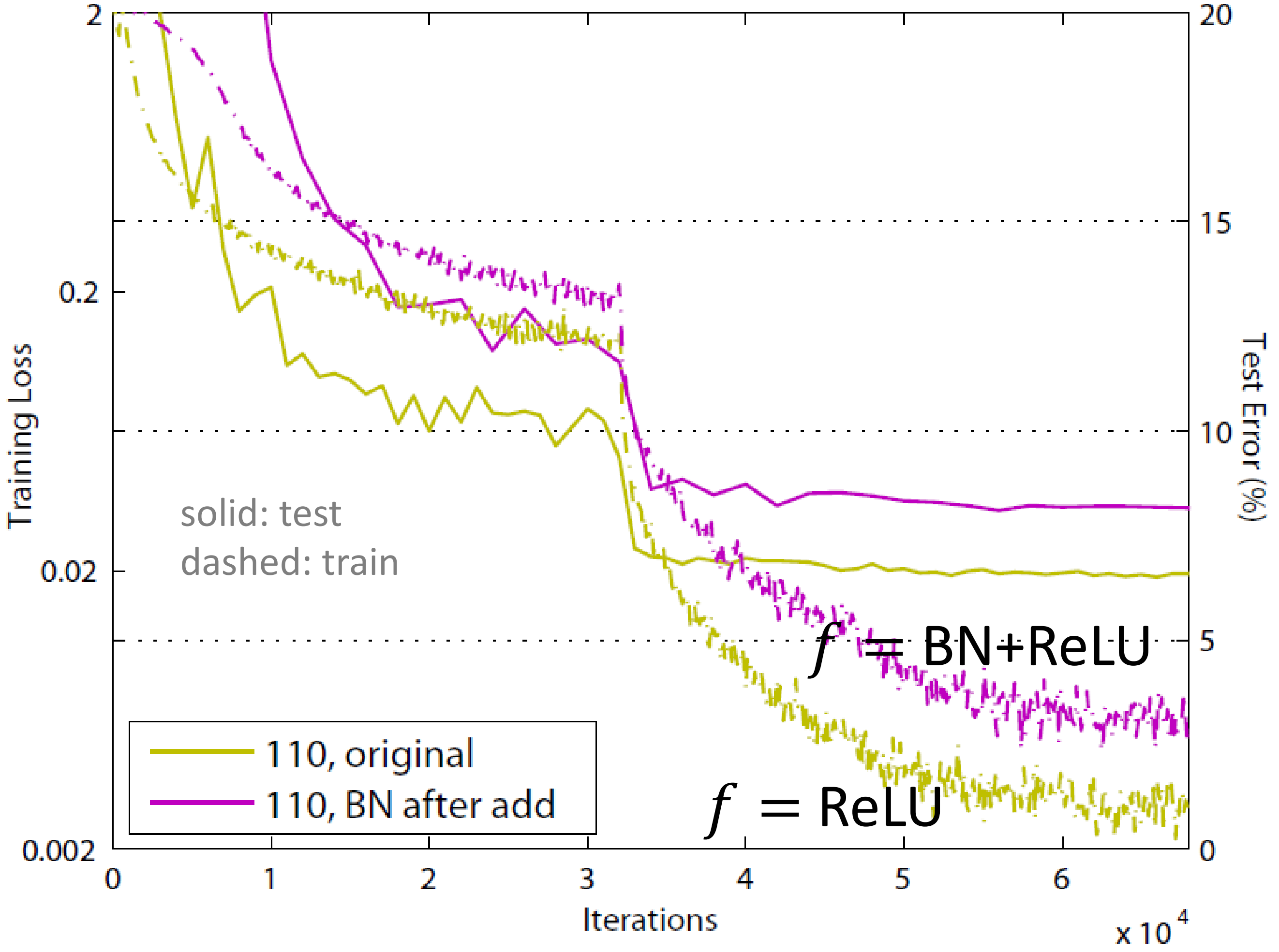


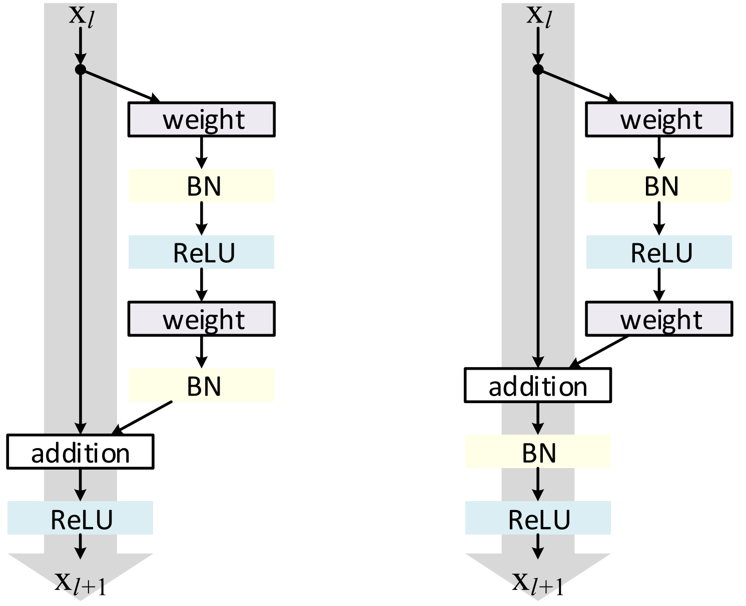
f is BN+ReLU



f is identity
(**pre-activation** ResNet)

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Identity Mappings in Deep Residual Networks”. arXiv 2016.



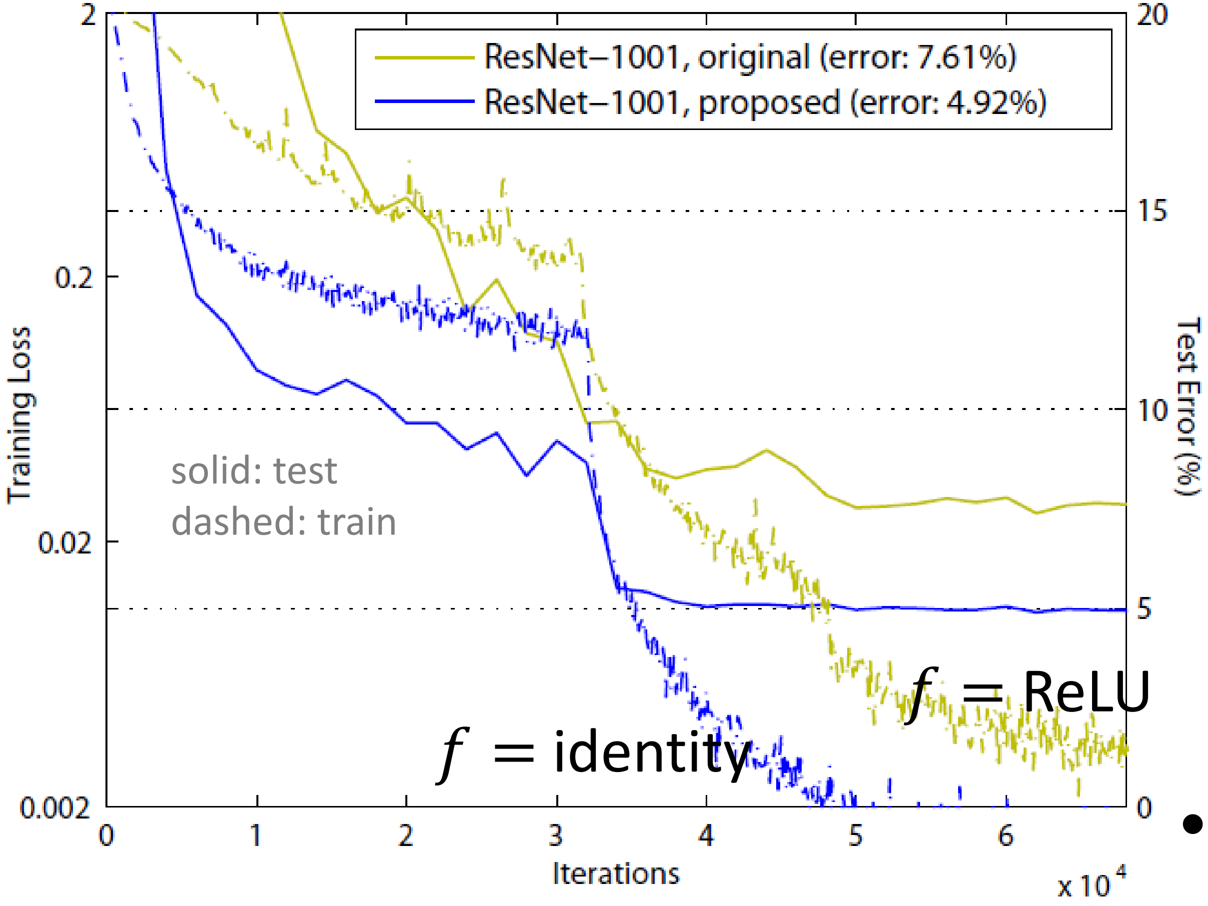


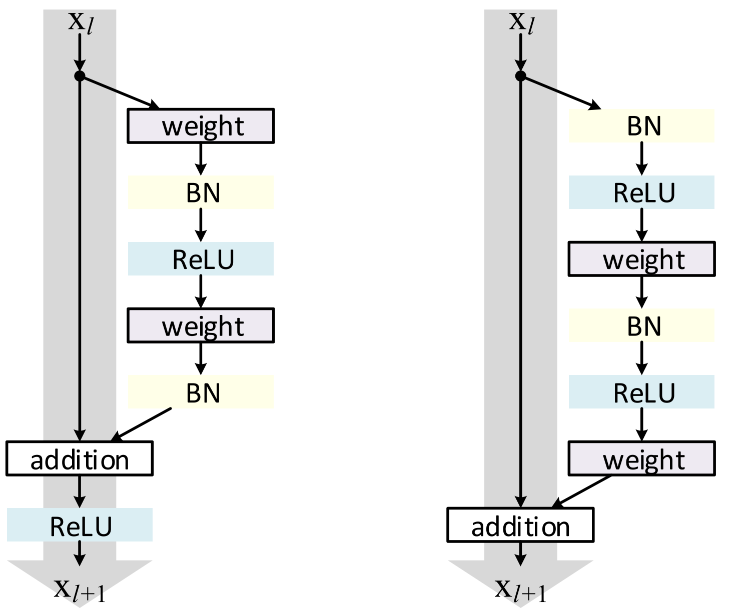
f = ReLU f = BN+ReLU

- BN could also block prop
- Keep the shortest pass as smooth as possible

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Identity Mappings in Deep Residual Networks”. arXiv 2016.

1001-layer ResNets on CIFAR-10





f = ReLU f = identity

- ReLU could also block prop when there are 1000 layers
- pre-activation design eases optimization (and improves generalization; see paper)

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Identity Mappings in Deep Residual Networks". arXiv 2016.

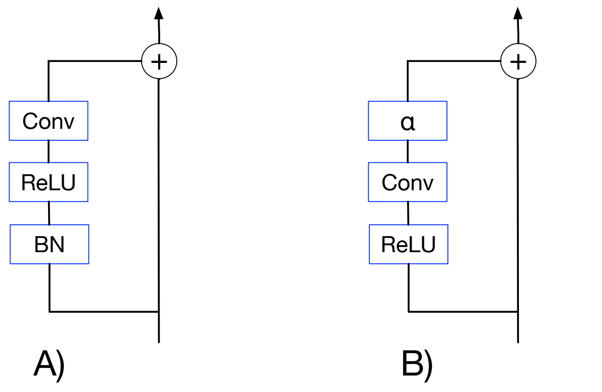
Batch Normalization Biases Residual Blocks Towards the Identity Function in Deep Networks

Soham De
DeepMind, London
sohamde@google.com

Samuel L. Smith
DeepMind, London
slsmith@google.com

https://arxiv.org/pdf/2002.10444.pdf

Residual block



w/BatchNorm w/SkipInit

- Batch normalization biases residual blocks towards the identity function

Table 1: Batch normalization enables us to train deep residual networks. We can recover this benefit without normalization if we introduce a scalar multiplier α on the end of the residual branch and initialize $\alpha = (1/\sqrt{d})$ or smaller (where d is the number of residual blocks). In practice, we advocate initializing $\alpha = 0$. We provide optimal test accuracies and optimal learning rates with error bars.

Depth	Test accuracy	Learning rate
Batch Normalization		
16	93.5 ± 0.1	2^{-1} (2^{-1} to 2^{-1})
100	94.7 ± 0.1	2^{-1} (2^{-2} to 2^{-0})
1000	94.6 ± 0.1	2^{-2} (2^{-3} to 2^{-0})
SkipInit ($\alpha = 1/\sqrt{d}$)		
16	93.0 ± 0.1	2^{-2} (2^{-2} to 2^{-1})
100	94.2 ± 0.1	2^{-1} (2^{-2} to 2^{-1})
1000	94.2 ± 0.0	2^{-1} (2^{-2} to 2^{-1})
SkipInit ($\alpha = 0$)		
16	93.3 ± 0.1	2^{-2} (2^{-2} to 2^{-2})
100	94.2 ± 0.1	2^{-2} (2^{-2} to 2^{-2})
1000	94.3 ± 0.2	2^{-2} (2^{-3} to 2^{-1})

Comparisons on CIFAR-10/100

CIFAR-10

method	error (%)
NIN	8.81
DSN	8.22
FitNet	8.39
Highway	7.72
ResNet-110 (1.7M)	6.61
ResNet-1202 (19.4M)	7.93
ResNet-164, pre-activation (1.7M)	5.46
ResNet-1001, pre-activation (10.2M)	**4.92** (4.89±0.14)

CIFAR-100

method	error (%)
NIN	35.68
DSN	34.57
FitNet	35.04
Highway	32.39
ResNet-164 (1.7M)	25.16
ResNet-1001 (10.2M)	27.82
ResNet-164, pre-activation (1.7M)	24.33
ResNet-1001, pre-activation (10.2M)	**22.71** (22.68±0.22)

*all based on moderate augmentation

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Identity Mappings in Deep Residual Networks”. arXiv 2016.

ImageNet Experiments

ImageNet single-crop (320x320) val error

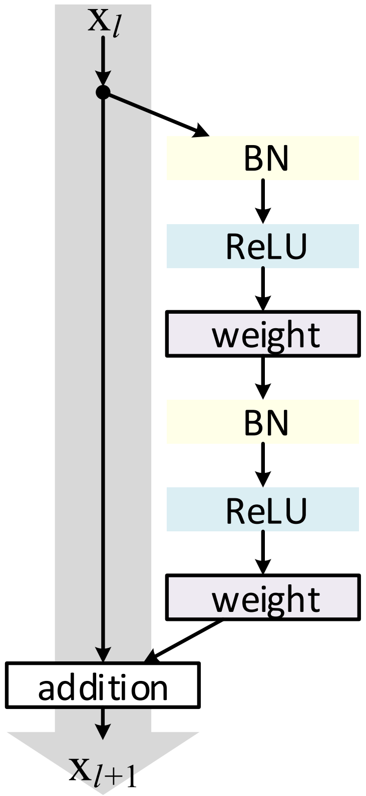
method	data augmentation	top-1 error (%)	top-5 error (%)
ResNet-152, original	scale	21.3	5.5
ResNet-152, pre-activation	scale	21.1	5.5
ResNet-200, original	scale	21.8	6.0
ResNet-200, pre-activation	scale	**20.7**	**5.3**
ResNet-200, pre-activation	scale + aspect ratio	**20.1***	**4.8***

*independently reproduced by:
https://github.com/facebook/fb.resnet.torch/tree/master/pretrained#notes
training code and models available.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Identity Mappings in Deep Residual Networks". arXiv 2016.

Summary of observations

- Keep the shortest path as smooth as possible
  - by making h and f identity
  - forward/backward signals directly flow through this path
- Features of any layers are additive outcomes
- 1000-layer ResNets can be easily trained and have better accuracy



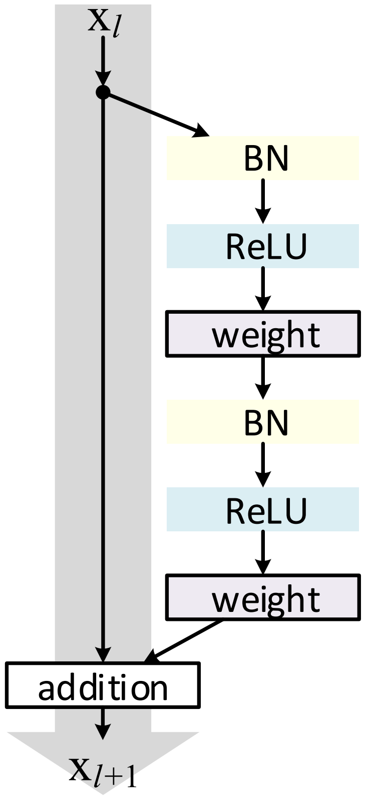
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Identity Mappings in Deep Residual Networks". arXiv 2016.

Future Works

- **Representation**
  - 1-layer block vs. multi-layer block?
  - Flat vs. Bottleneck?
  - Inception-ResNet [Szegedy et al 2016]
  - ResNet in ResNet [Targ et al 2016]
  - Width vs. Depth [Zagoruyko & Komodakis 2016]

- **Generalization**
  - DropOut, MaxOut, DropConnect, …
  - Drop Layer (Stochastic Depth) [Huang et al 2016]

- **Optimization**
  - Without residual?



Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Identity Mappings in Deep Residual Networks”. arXiv 2016.

More Visual Recognition Tasks

ResNet-based methods lead on these benchmarks (incomplete list):

- ImageNet classification, detection, localization
- MS COCO detection, segmentation
- PASCAL VOC detection, segmentation

- MPII Human pose estimation [Newell et al 2016]
- Depth estimation [Laina et al 2016]
- Segment proposal [Pinheiro et al 2016]
- …

	mean	aero plane	bicycle	bird	boat	bottle	bus	car
DeepLabv2-CRF [?]	79.7	92.6	60.4	91.6	63.4	76.3	95.0	88.4
CASIA_SegResNet_CRF_COCO [?]	79.3	93.8	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Adelaide_VeryDeep_FCN_VOC [?]	79.1	91.9	48.1	93.4	69.3	75.5	94.2	87.5
LRR_4x_COCO [?]	78.7	93.2	44.2	89.4	65.4	74.9	93.9	87.0
CASIA_IVA_OASeg [?]	78.3	93.8	41.9	89.4	67.5	71.5	94.6	85.3
Oxford_TVG_HO_CRF [?]	77.9	92.5	59.1	90.3	70.6	74.4	92.4	84.1
Adelaide_Context_CNN_CRF_COCO [?]	77.8	92.9	39.6	84.0	67.9	75.3	92.7	83.8

ResNet-101

PASCAL **segmentation** leaderboard

	mean	aero plane	bicycle	bird	boat	bottle	bus	car	cat
Faster RCNN, ResNet (VOC+COCO) [?]	83.8	92.1	88.4	84.8	75.9	71.4	86.3	87.8	94.2
R-FCN, ResNet (VOC+COCO) [?]	82.0	89.5	88.3	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
OHEM+FRCN, VGG16, VOC+COCO [?]	80.1	90.1	87.4	79.9	65.8	66.3	86.1	85.0	92.9
SSD500 VGG16 VOC + COCO [?]	78.7	89.1	85.7	78.9	63.3	57.0	85.3	84.1	92.3
HFM_VGG16 [?]	77.5	88.8	85.1	76.8	64.8	61.4	85.0	84.1	90.0
IFRN_07+12 [?]	76.6	87.8	83.9	79.0	64.5	58.9	82.2	82.0	91.4
ION [?]	76.4	87.5	84.7	76.8	63.8	58.3	82.6	79.0	90.9

ResNet-101

PASCAL **detection** leaderboard

Potential Applications

ResNets have shown outstanding or promising results on:

- Visual Recognition
- Image Generation (Pixel RNN, Neural Art, etc.)
- Natural Language Processing (Very deep CNN)
- Speech Recognition (preliminary results)
- Advertising, user prediction (preliminary results)

Conclusions of the Tutorial

- Deep Residual Learning:
  - Ultra deep networks can be easy to train
  - Ultra deep networks can gain accuracy from depth
  - Ultra deep representations are well transferrable
  - Now 200 layers on ImageNet and 1000 layers on CIFAR!

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Identity Mappings in Deep Residual Networks”. arXiv 2016.

Resources

Thank You!
Q & A

- Models and Code
  - Our ImageNet models in Caffe: https://github.com/KaimingHe/deep-residual-networks

- Many available implementation
  (see https://github.com/KaimingHe/deep-residual-networks)
  - Facebook AI Research's Torch ResNet: https://github.com/facebook/fb.resnet.torch
  - Torch, CIFAR-10, with ResNet-20 to ResNet-110, training code, and curves: code
  - Lasagne, CIFAR-10, with ResNet-32 and ResNet-56 and training code: code
  - Neon, CIFAR-10, with pre-trained ResNet-32 to ResNet-110 models, training code, and curves: code
  - Torch, MNIST, 100 layers: blog, code
  - A winning entry in Kaggle's right whale recognition challenge: blog, code
  - Neon, Place2 (mini), 40 layers: blog, code
  - …….

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Identity Mappings in Deep Residual Networks". arXiv 2016.

Exploring the Limits of Weakly Supervised Pretraining

Laurens van der Maaten

ECCV 2018

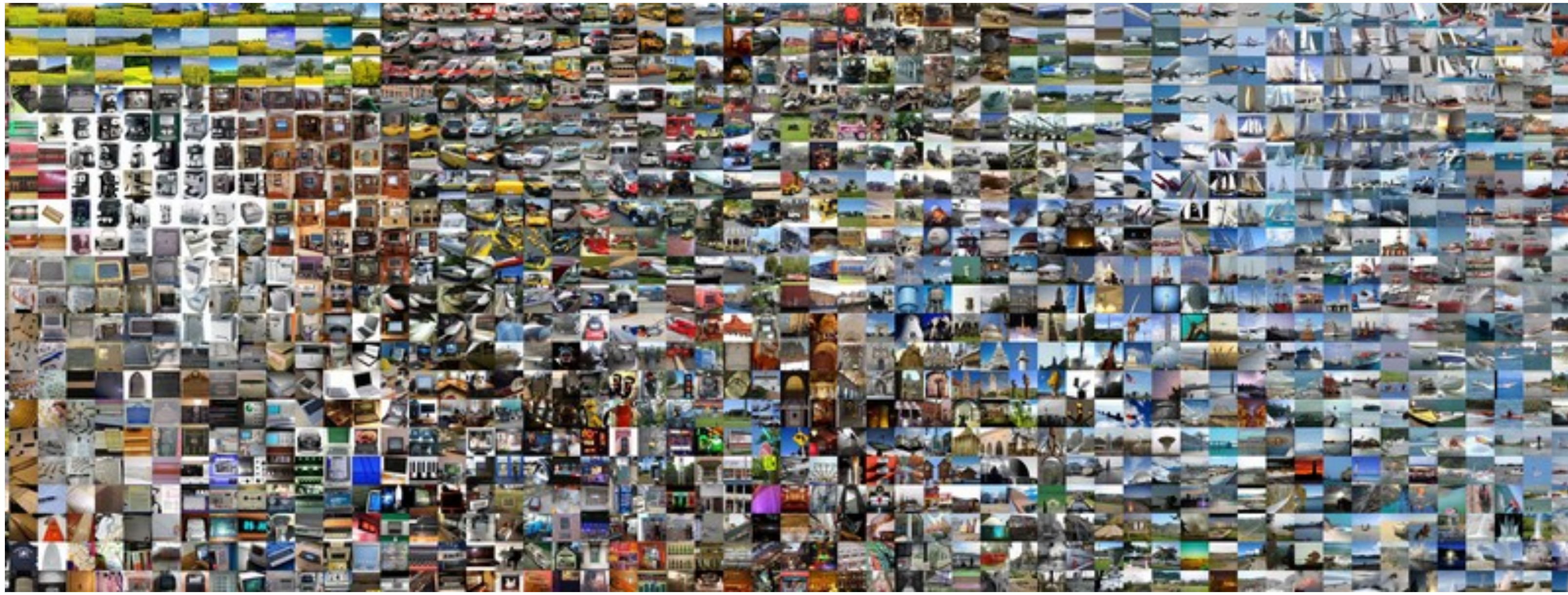
Dhruv Mahajan Ross Girshick Vignesh Ramanathan Kaiming He Manohar Paluri Yixuan Li Ashwin Bharambe

facebook
Artificial Intelligence Research

https://arxiv.org/pdf/1805.00932.pdf

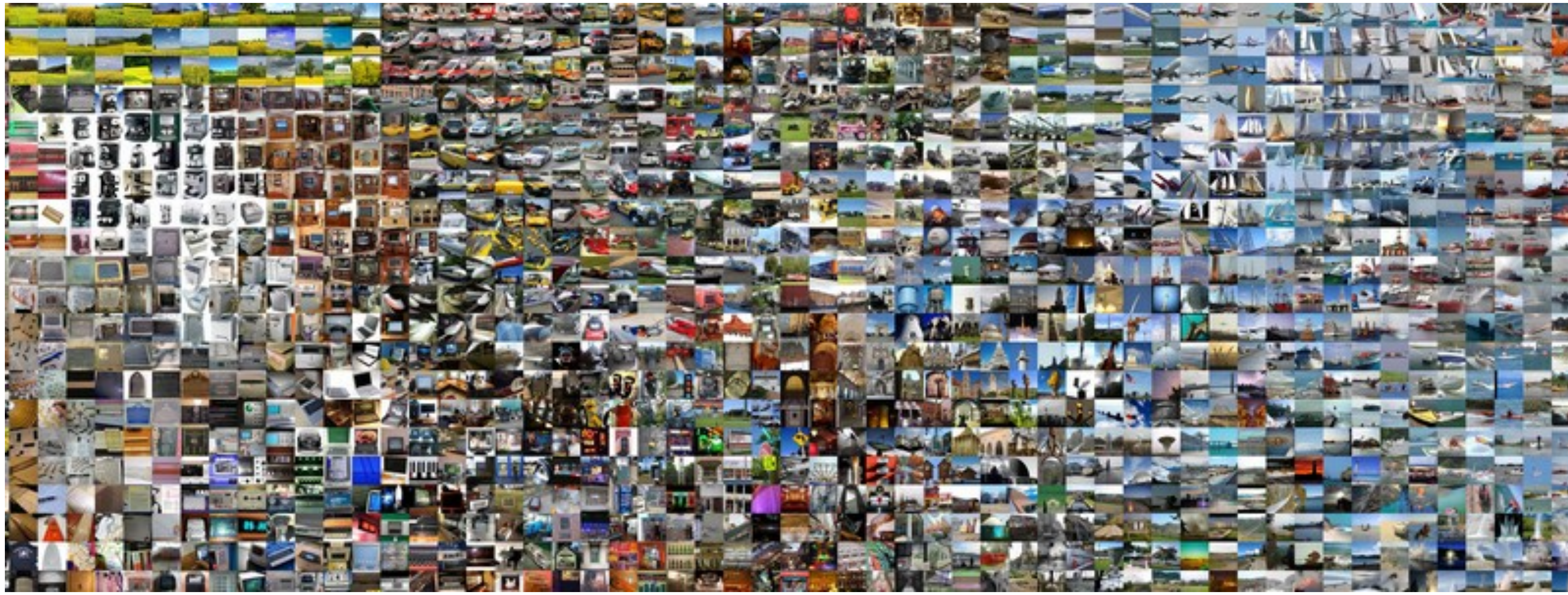
Pretraining Vision Models

- First, train a model on a large "source" dataset (say, ImageNet)



Pretraining Vision Models

- First, train a model on a large "source" dataset (say, ImageNet)



- Finetune on a small "target" dataset
- Measure accuracy on target task

Research question

Can we use large amounts of weakly supervised images for pretraining?

Highlights

- We pretrain models by predicting relevant hashtags for images
- We pretrain models to predict 17.5K hashtags for 3.5B images
- After finetuning, we beat the state-of-the-art on, *e.g.*, ImageNet

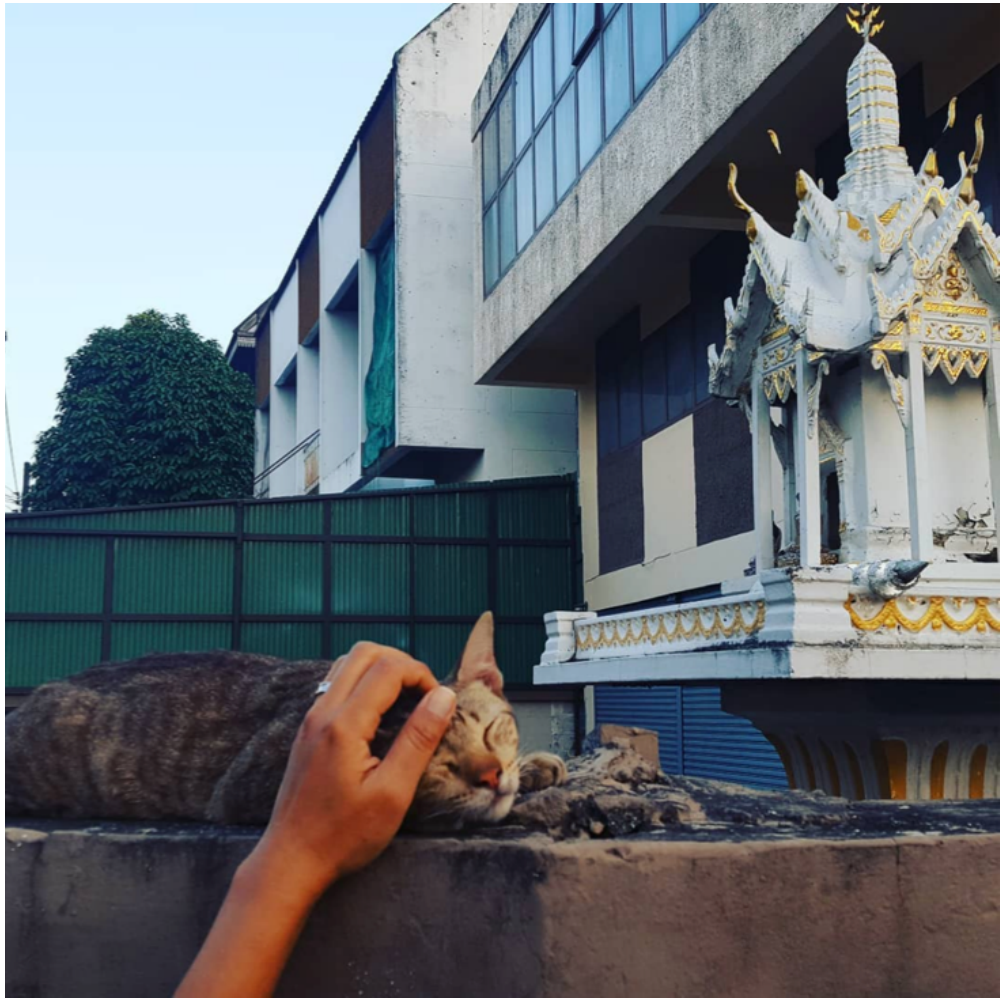
Hashtag Supervision

- It is easy to get billions of public images and hashtags
- Hashtags are more structured than captions
- Hashtags were often assigned to make images "searchable"

#cheesecake #birthday

Hashtag Supervision

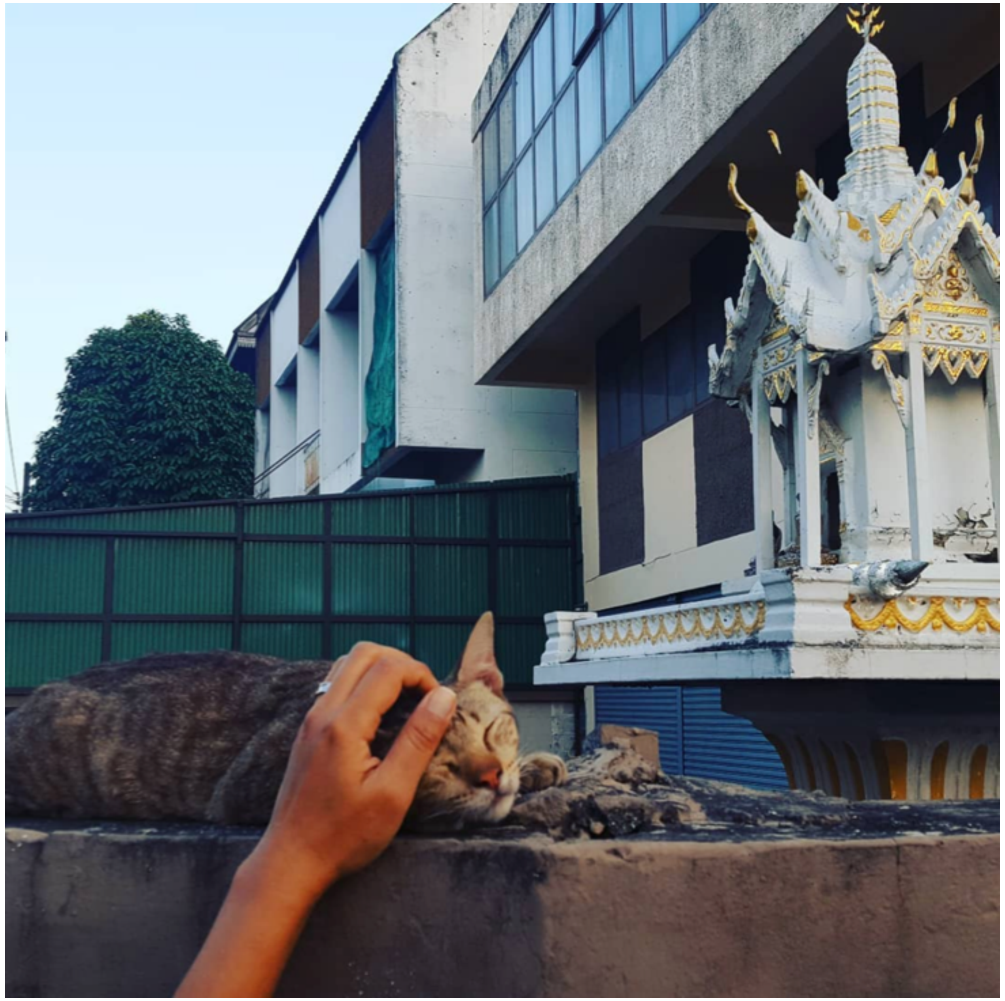
- But hashtags are not perfect supervision



#cat #travel #thailand #family

Hashtag Supervision

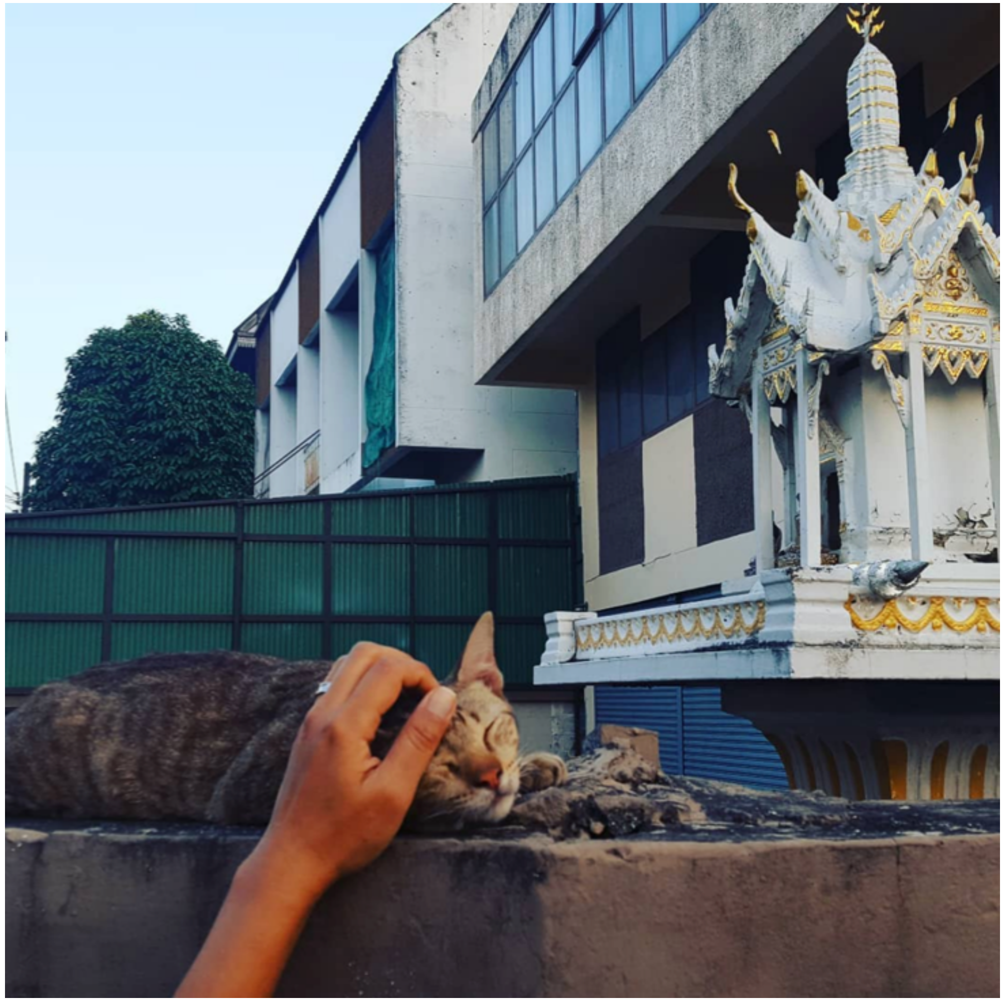
- But hashtags are not perfect supervision
- Some hashtags are not visually relevant



#cat #travel #thailand #family

Hashtag Supervision

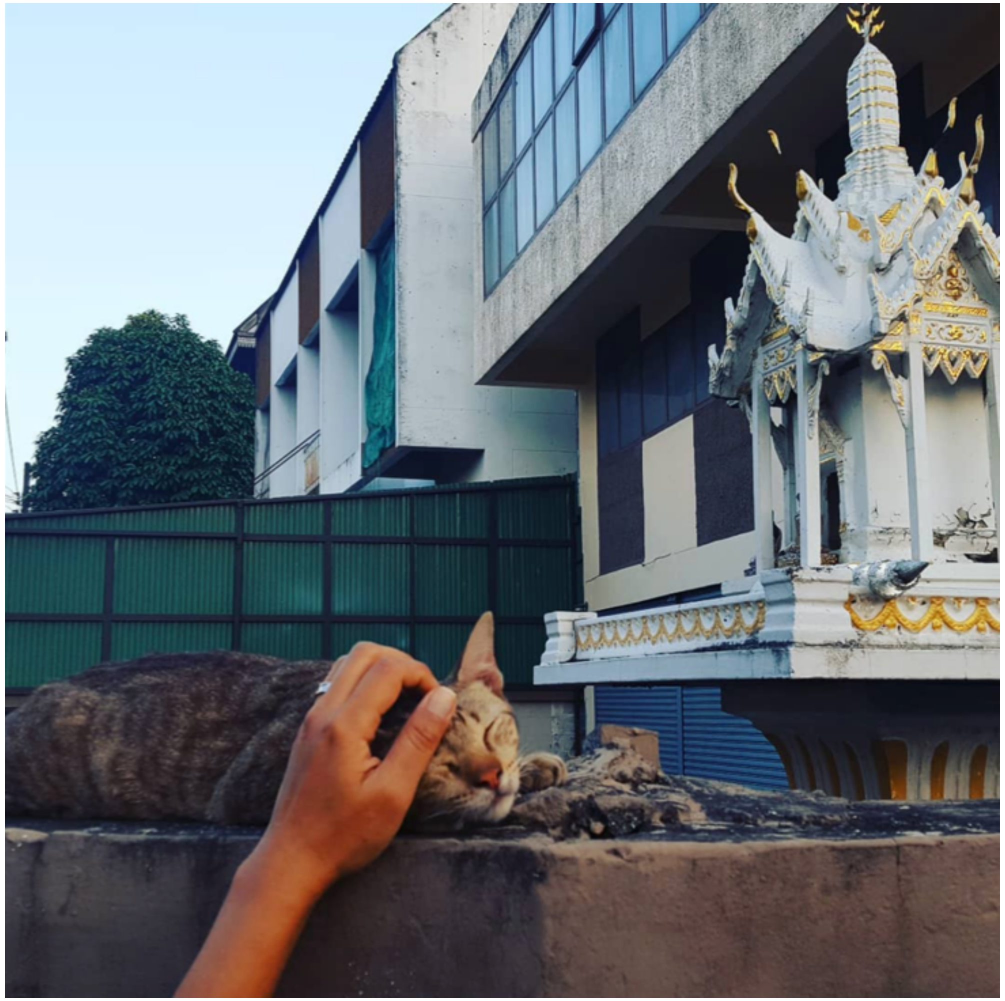
- But hashtags are not perfect supervision
- Some hashtags are not visually relevant
- Other hashtags are not in the photo



#cat #travel #thailand #family

Hashtag Supervision

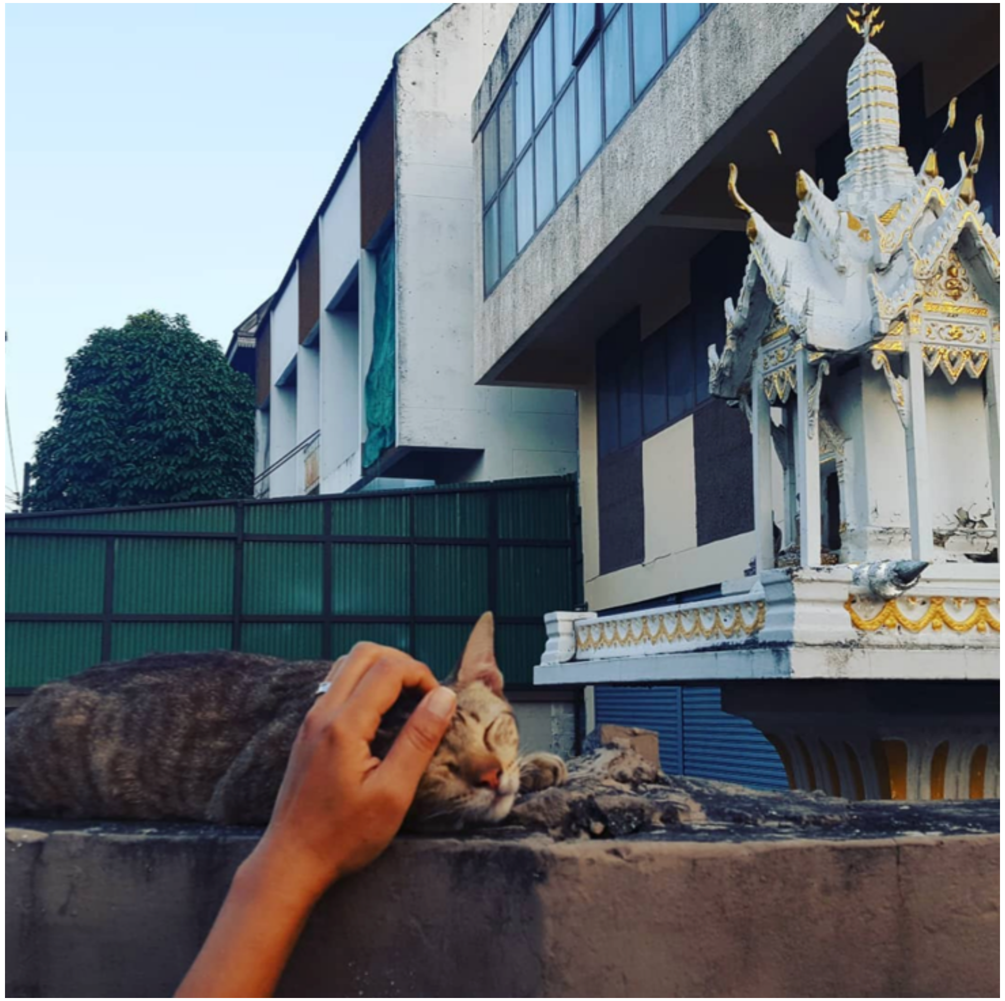
- But hashtags are not perfect supervision
- Some hashtags are not visually relevant
- Other hashtags are not in the photo
- And there are many false negatives



#cat #travel #thailand #family
#building #fence #...

Hashtag Supervision

- But hashtags are not perfect supervision
- Some hashtags are not visually relevant
- Other hashtags are not in the photo
- And there are many false negatives
- Is this noise bias or variance? Is scaling up sufficient to reduce the variance?



#cat #travel #thailand #family
#building #fence #...

Experiments

- Select a set of hashtags
- Download all public Instagram images that has at least one of these hashtags
- Use WordNet synsets to merge hashtags into canonical form (merge #brownbear and #ursusarctos)

Experiments

- Select a set of hashtags
- Download all public Instagram images that has at least one of these hashtags
- Use WordNet synsets to merge hashtags into canonical form (merge #brownbear and #ursusarctos)
- The final list has 17,517 hashtags

1 aar
2 aardvark
3 aardwolf
4 aba
5 abaca
6 abacus
7 abalone
8 abatis
9 abaya
10 abbey
11 abele
12 abelia
13 abies
14 abila
15 abm
16 abortus
17 abronia
18 absinth
19 absinthe
20 abstraction
21 abstractionism
22 abutilon
23 abutment
24 abyss
25 abyssinian
26 acacia
27 acaciadealbata
28 academy
29 acalypha
30 acanthaceae
31 acanthurus
32 acanthus
33 acanthusmollis
34 acapulcogold
35 acarus
36 accelerator
37 accelerometer
38 access
39 accessory
40 accident
41 accipiter
42 accipiternisus
43 accipitridae
44 accommodation
45 accompaniment
46 accordion
47 accoutrement
48 accumulator
49 ace
50 aceofclubs
51 aceofdiamonds
52 aceofhearts
53 aceofspades
54 acer
55 acerjaponicum
56 acerola
57 acerpalmatum
58 acerrubrum
59 acetaminophen
60 acetate
61 acheron
62 acherontia
63 acherontiaatropos
64 achillea
65 achilleamillefolium
66 achimenes
67 acid
68 acidophilus
69 acinonyxjubatus
70 acinus
71 ackee
72 aconcagua
73 aconite
74 aconitum
75 acorn
76 acornsquash
77 acousticguitar
78 acoustics
79 acrididae
80 acrobates
81 acropolis
82 acropora
83 acrylic
84 acrylicpaints
85 actias
86 actiasluna

...

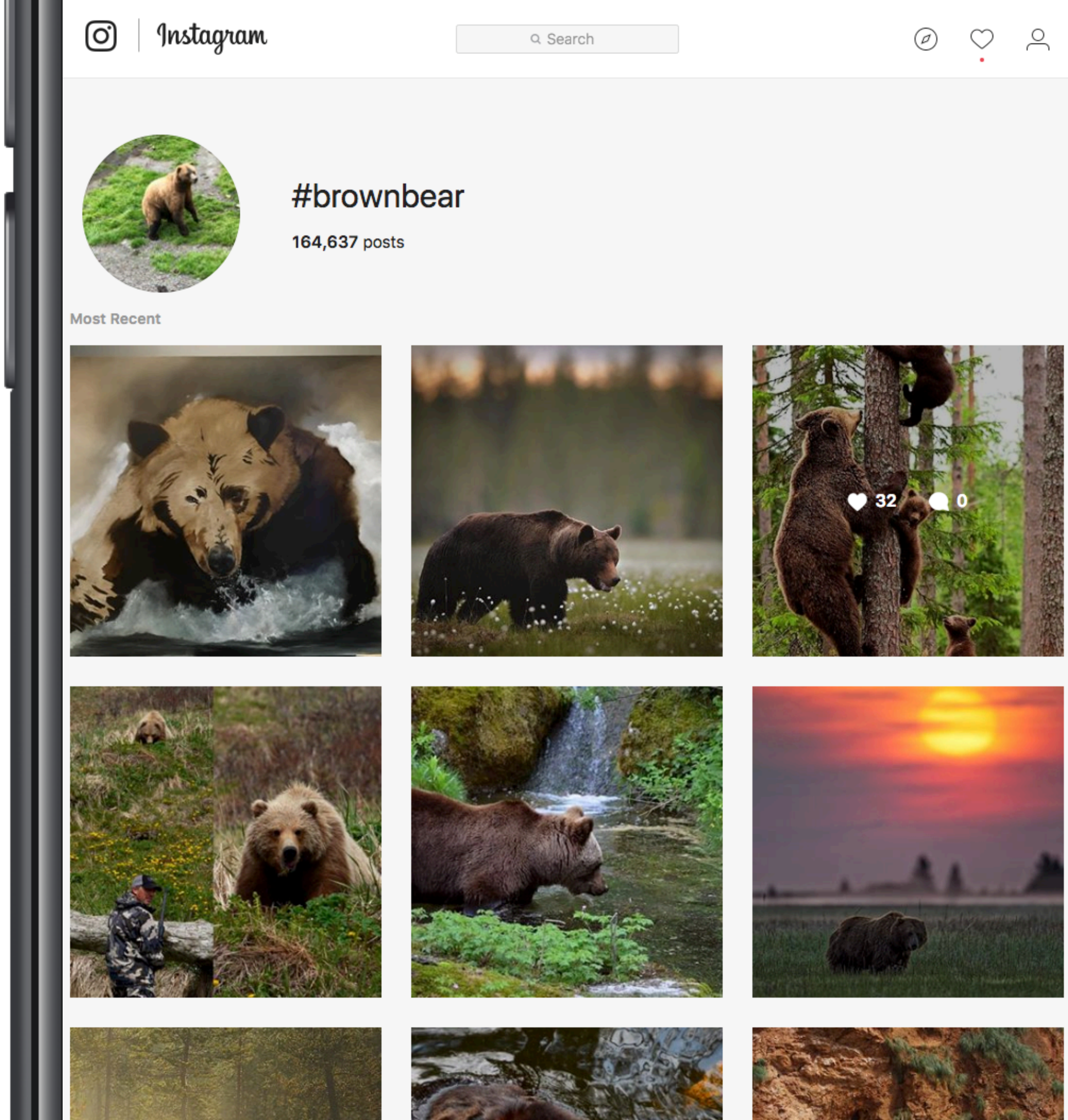
17474 yurt
17475 zabaglione
17476 zambeziriver
17477 zamboni
17478 zamia
17479 zantac
17480 zantedeschia
17481 zap
17482 zapper
17483 zarf
17484 zea
17485 zebra
17486 zebrafinch
17487 zebrawood
17488 zebu
17489 zero
17490 zeus
17491 zhujiang
17492 ziggurat
17493 zill
17494 zimmerframe
17495 zinfandel
17496 zing
17497 zingiber
17498 zinnia
17499 zipgun
17500 zipper
17501 zither
17502 ziti
17503 ziziphus
17504 zizz
17505 zodiac
17506 zoloft
17507 zombi
17508 zoologicalgarden
17509 zoom
17510 zooplankton
17511 zootsuit
17512 zori
17513 zoysia
17514 zuiderzee
17515 zygnema
17516 zygocactus
17517 zygoptera

Experiments

- Select a set of hashtags
- Download all public Instagram images that has at least one of these hashtags
- Use WordNet synsets to merge hashtags into canonical form (merge #brownbear and #ursusarctos)
- Final dataset has ~3.5 **billion** images

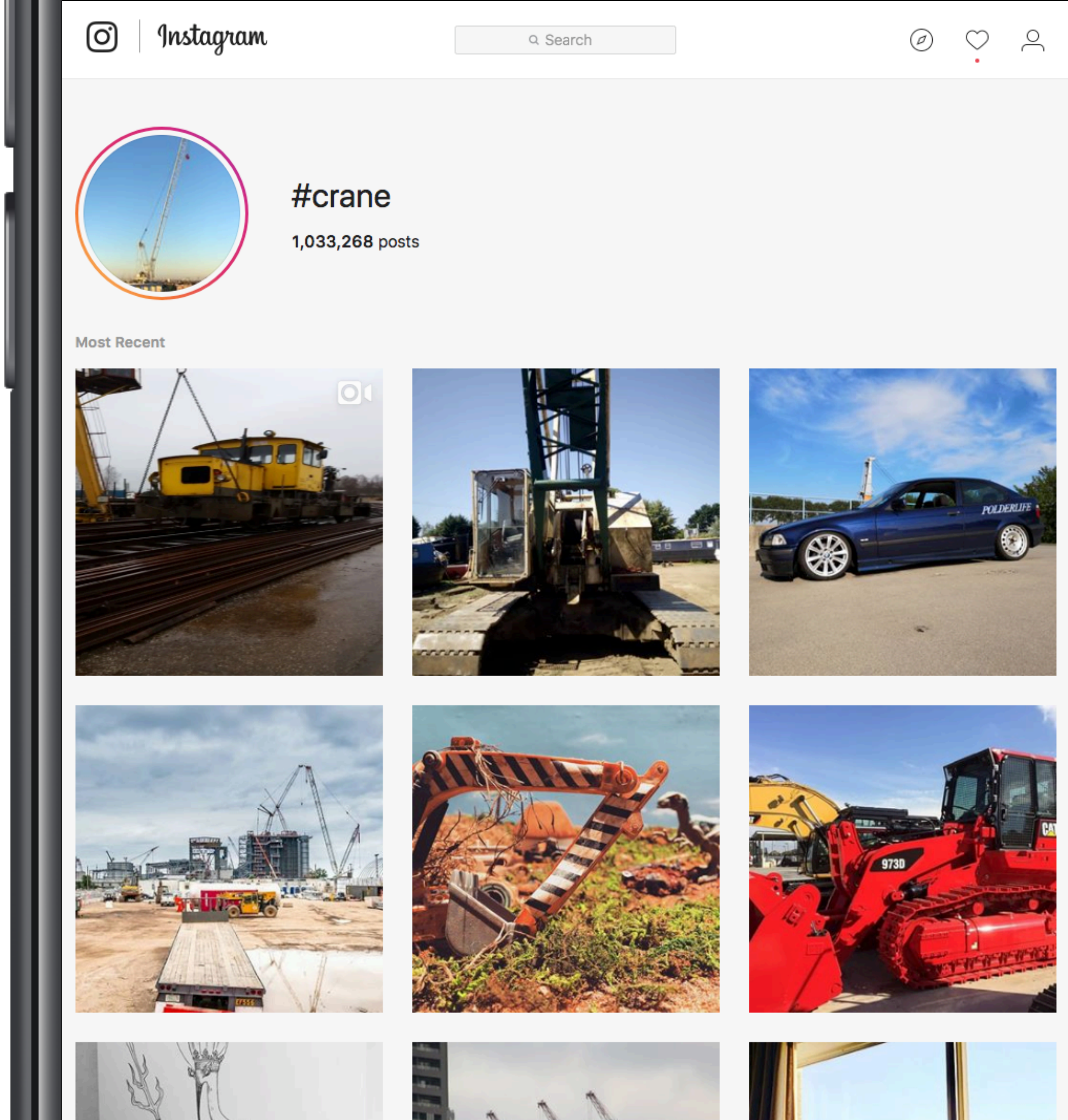
Experiments

- Select a set of hashtags
- Download all public Instagram images that has at least one of these hashtags
- Use WordNet synsets to merge hashtags into canonical form (merge #brownbear and #ursusarctos)
- Final dataset has ~3.5 **billion** images



Experiments

- Select a set of hashtags
- Download all public Instagram images that has at least one of these hashtags
- Use WordNet synsets to merge hashtags into canonical form (merge #brownbear and #ursusarctos)
- Final dataset has ~3.5 **billion** images



Experiments

- Select a set of hashtags
- Download all public Instagram images that has at least one of these hashtags
- Use WordNet synsets to merge hashtags into canonical form (merge `#brownbear` and `#ursusarctos`)
- De-duplicate test sets against Instagram!

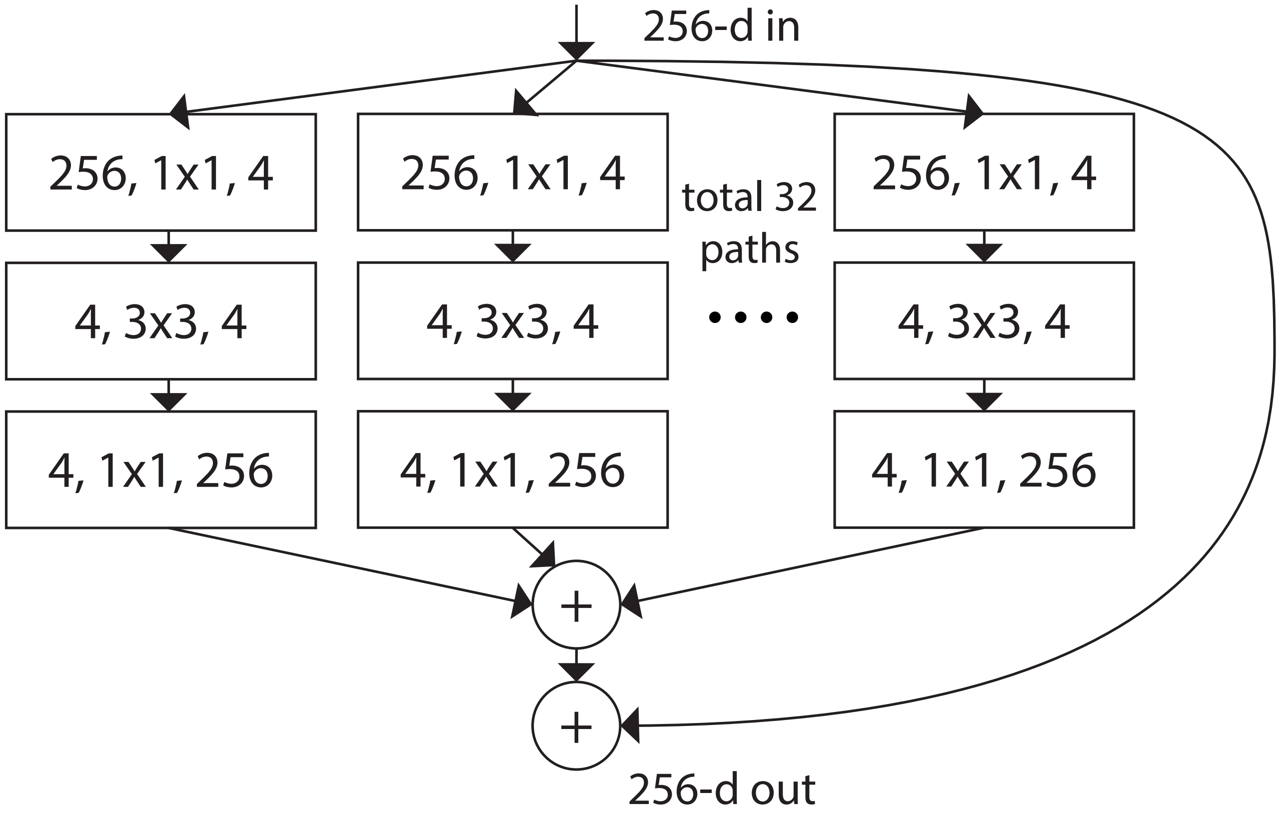
We developed strong near-duplicate detector:

We found that <0.3% of ImageNet images are in our 3.5B Instagram sample.

(This is actually a lower percentage than in most prior papers on "transfer" learning.)

Experiments

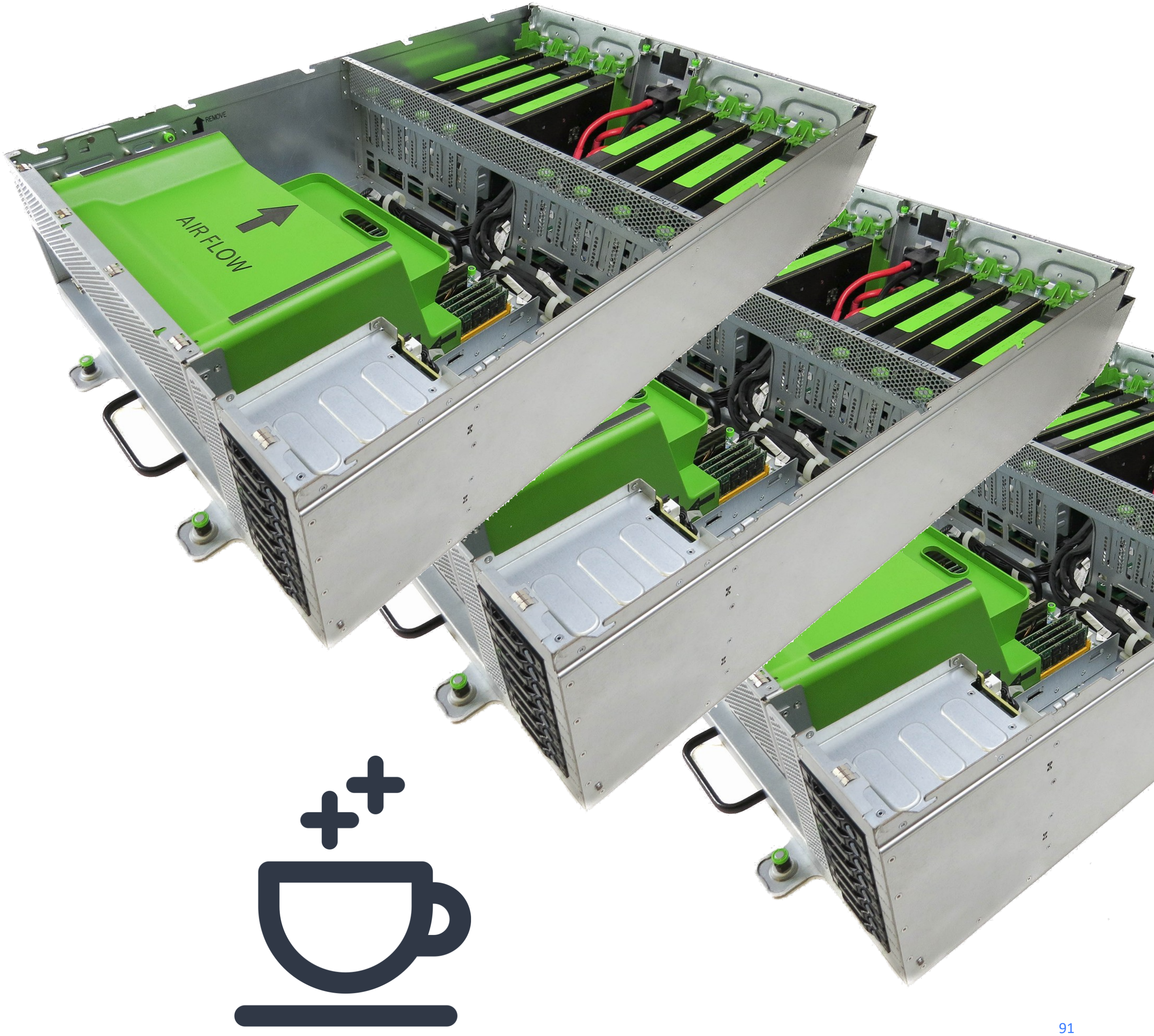
- Train ResNeXt-32x*C*d convolutional networks
- Use *c*-of-*K* vector to represent multiple labels
- Train to minimize multi-class logistic loss



most experiments use ResNeXt-101 32x16d

Experiments

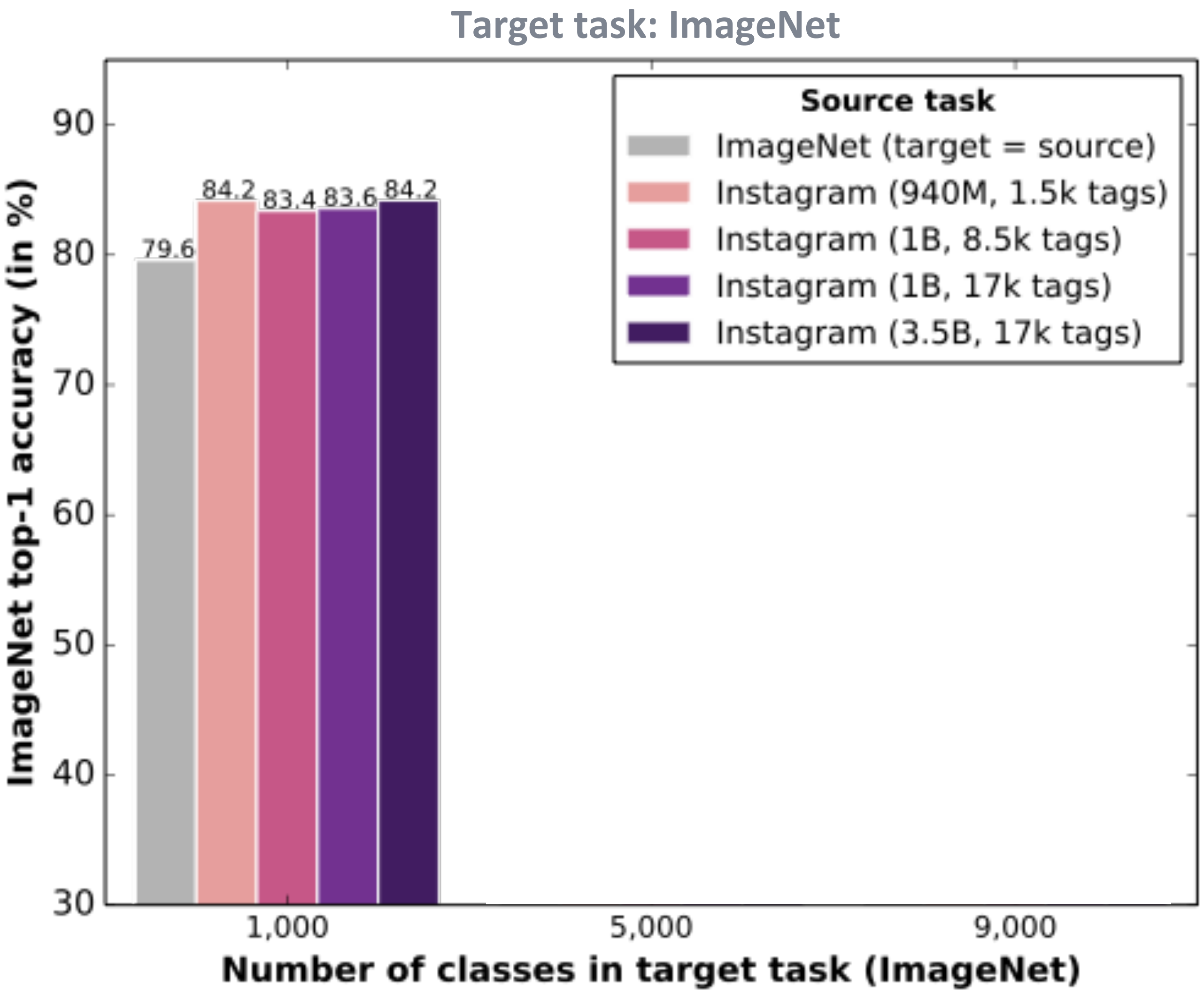
- Train ResNeXt-32x*C*d convolutional networks
- Use *c*-of-*K* vector to represent multiple labels
- Train to minimize multi-class logistic loss
- Distribute training batches across 336 GPUs
- Scale learning rate by batch size *(N=8,064)* after learning rate "warm-up" (Goyal *et al.*, 2017)



Results

Fix Model; Vary Data

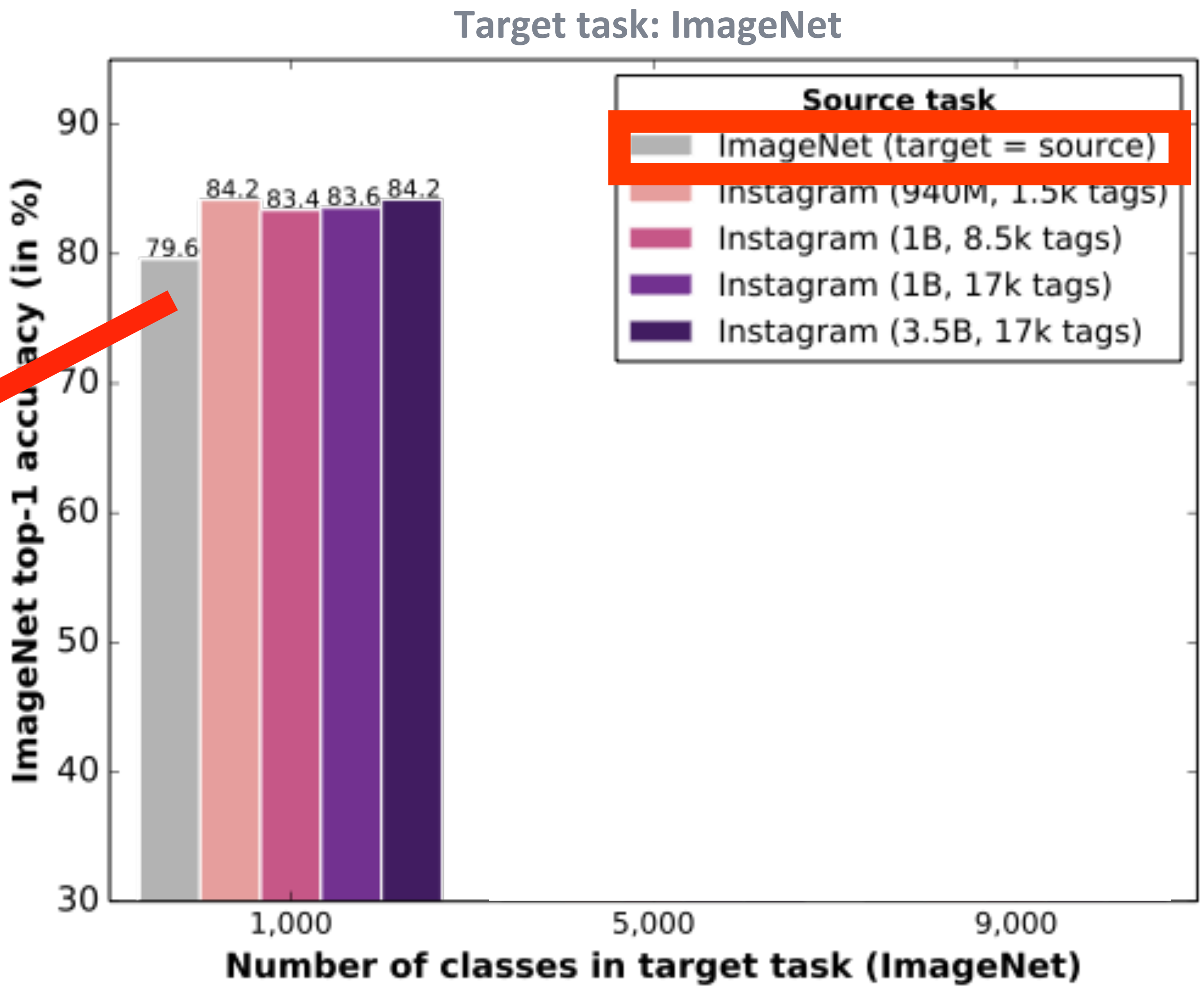
- Pretrain model on ImageNet or Instagram
- Finetune on ImageNet



Fix Model; Vary Data

- Pretrain model on ImageNet or Instagram
- Finetune on ImageNet

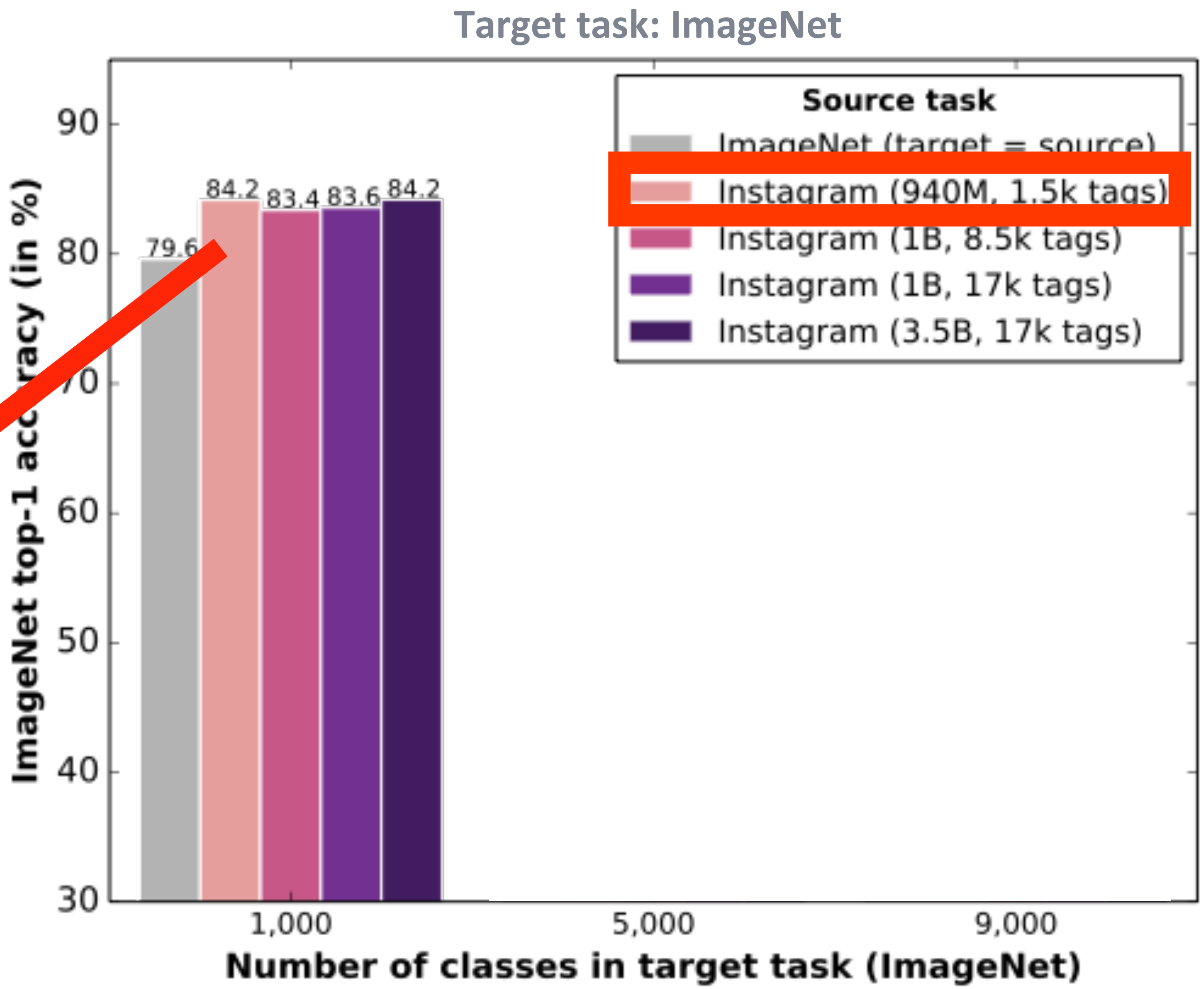
"standard" ImageNet training



Fix Model;
Vary Data

- Pretrain model on ImageNet or Instagram
- Finetune on ImageNet

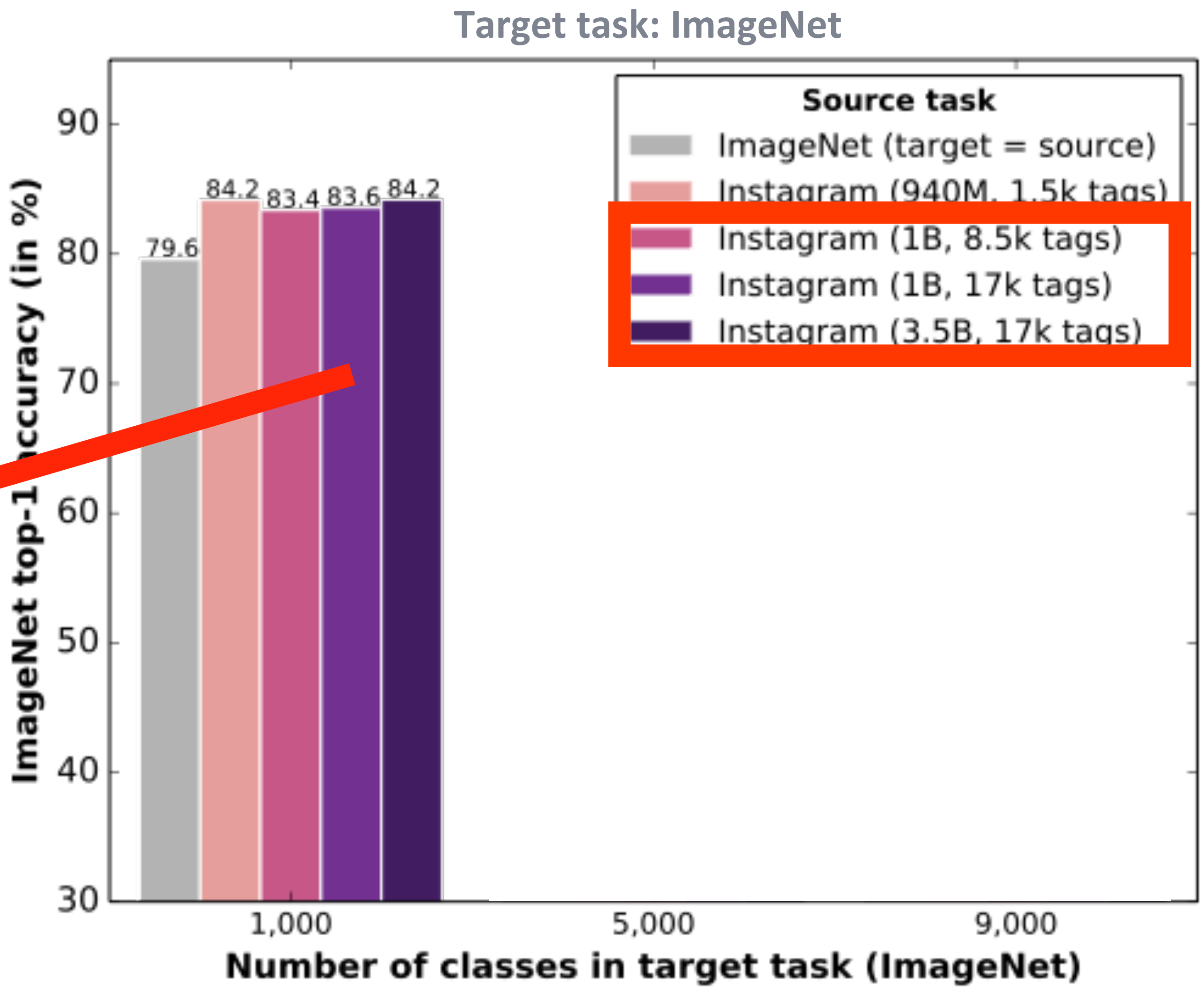
pre-training on 1B Instagram images, selected to match ImageNet classes



Fix Model;
Vary Data

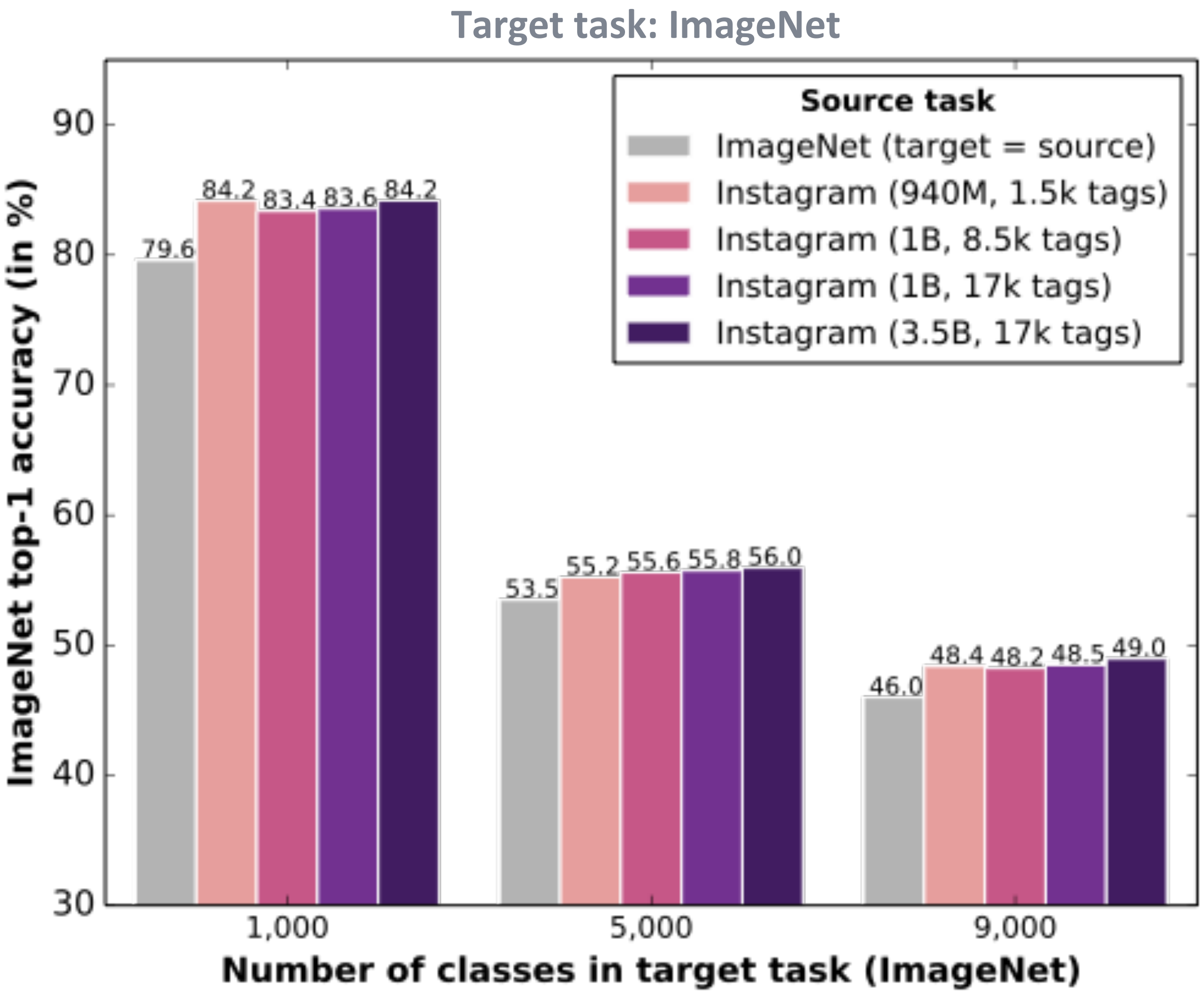
- Pretrain model on ImageNet or Instagram
- Finetune on ImageNet

pretraining on 1-3.5B Instagram images, without selection



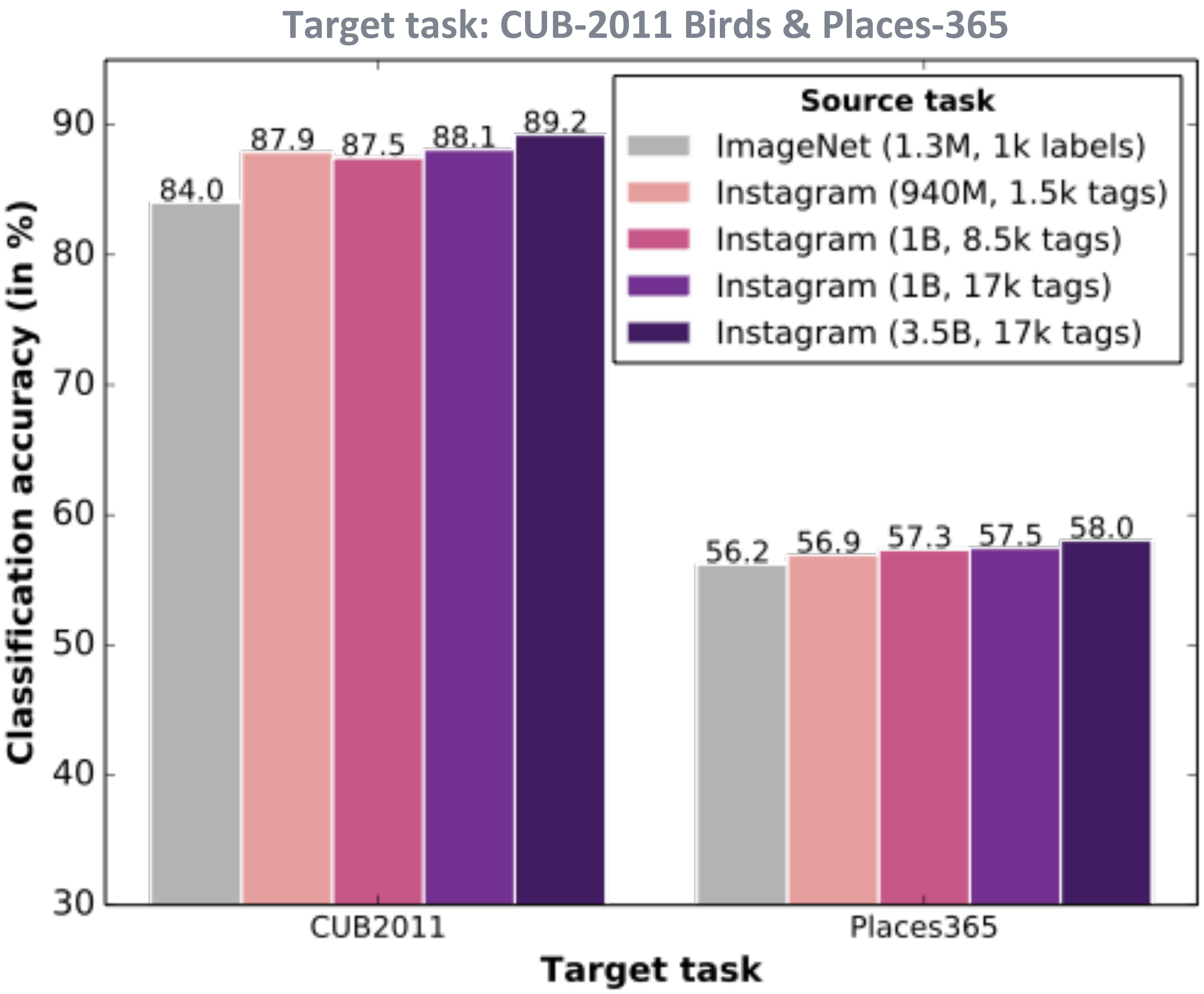
Fix Model; Vary Data

- Pretrain model on ImageNet or Instagram
- Finetune on ImageNet
- Similar results on larger versions of ImageNet



Fix Model; Vary Data

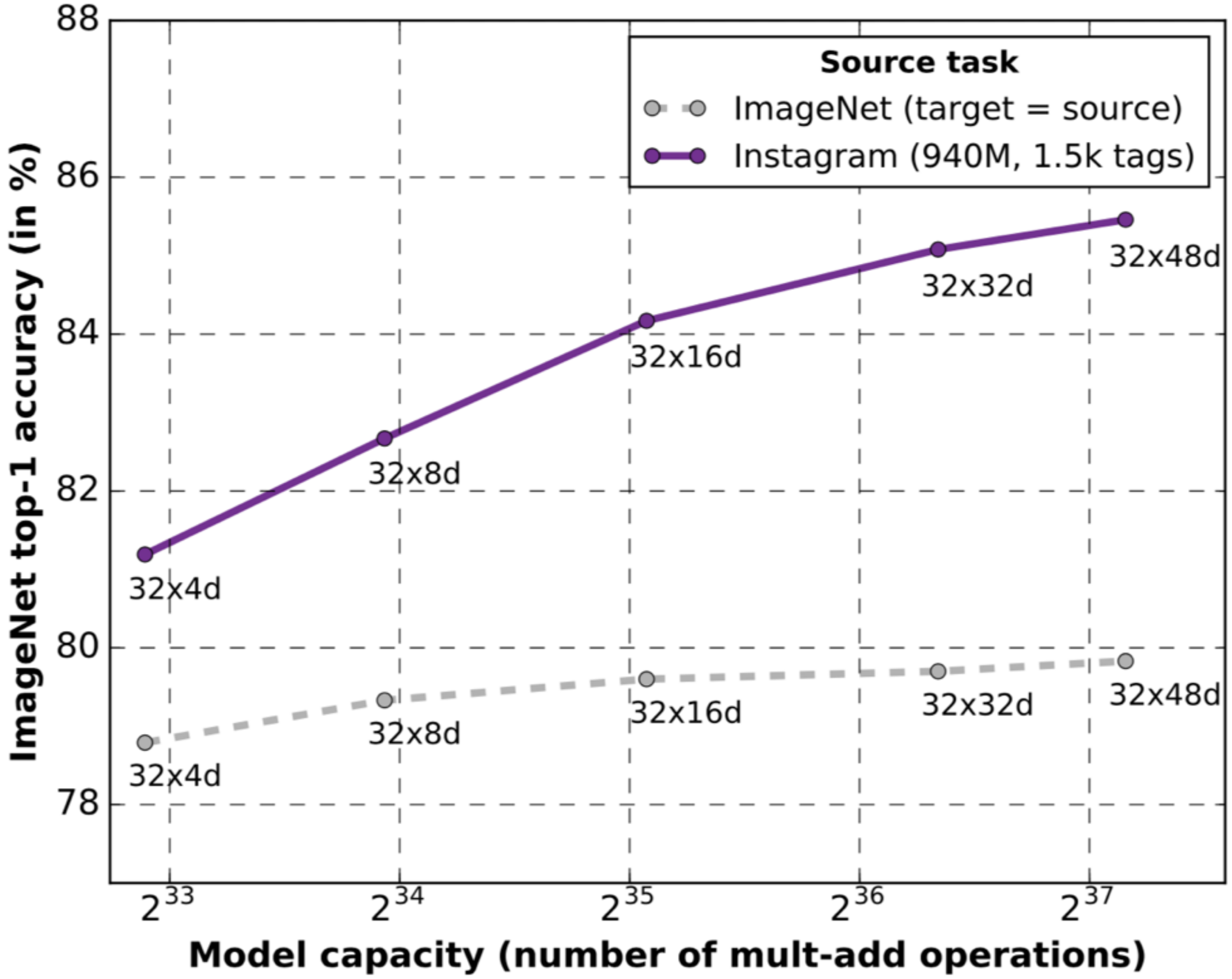
- We observe similar results on the CUB-2011 Birds dataset and Places-365



Fix Data;
Vary Model

- Increasing model capacity has a larger positive effect

- Even lower error rates may be possible?



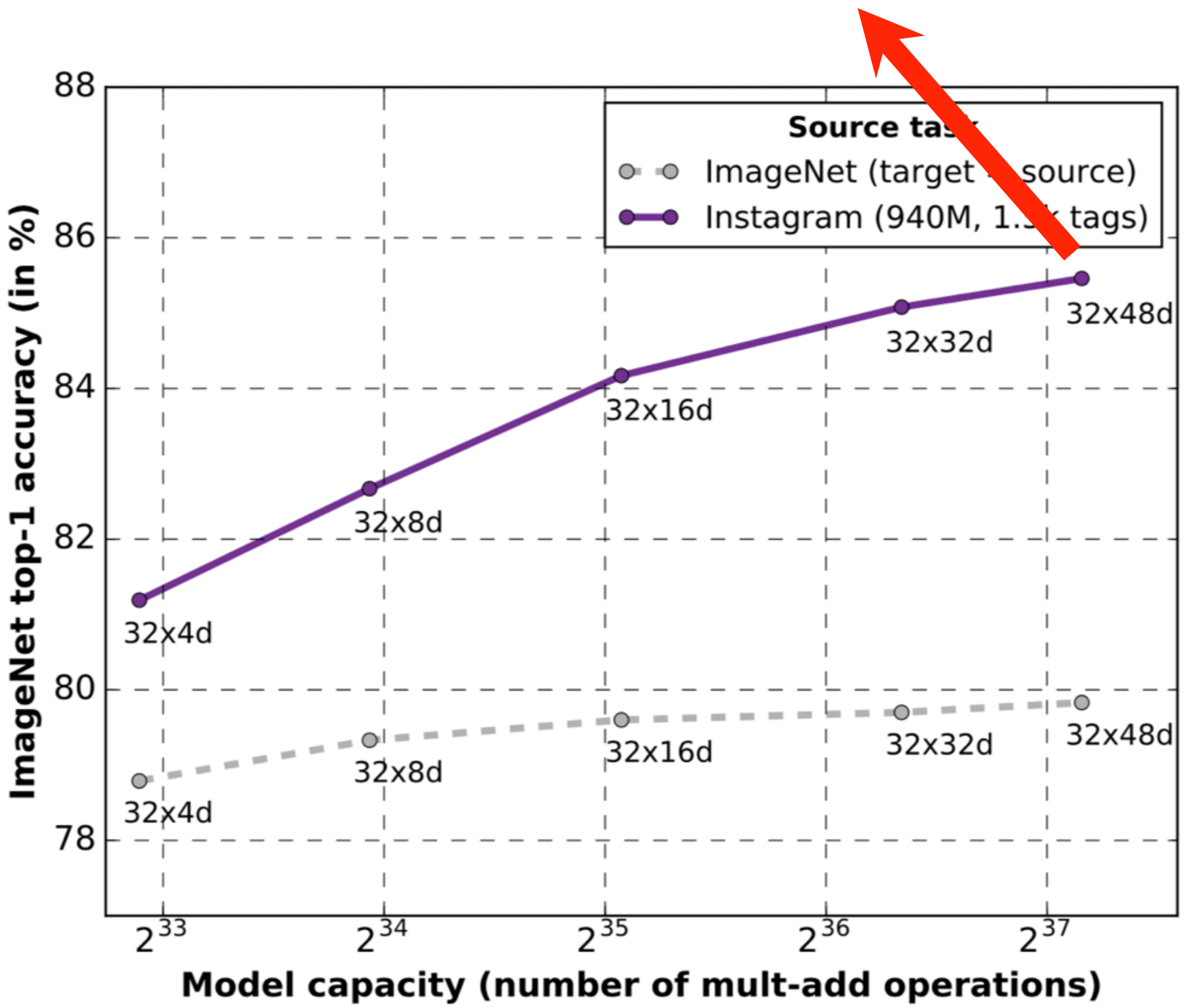
facebook
Artificial Intelligence Research

Fix Data;
Vary Model

- Increasing model capacity has a larger positive effect

- Even lower error rates may be possible?

best result: 85.4% top-1 accuracy



State-of-the-art

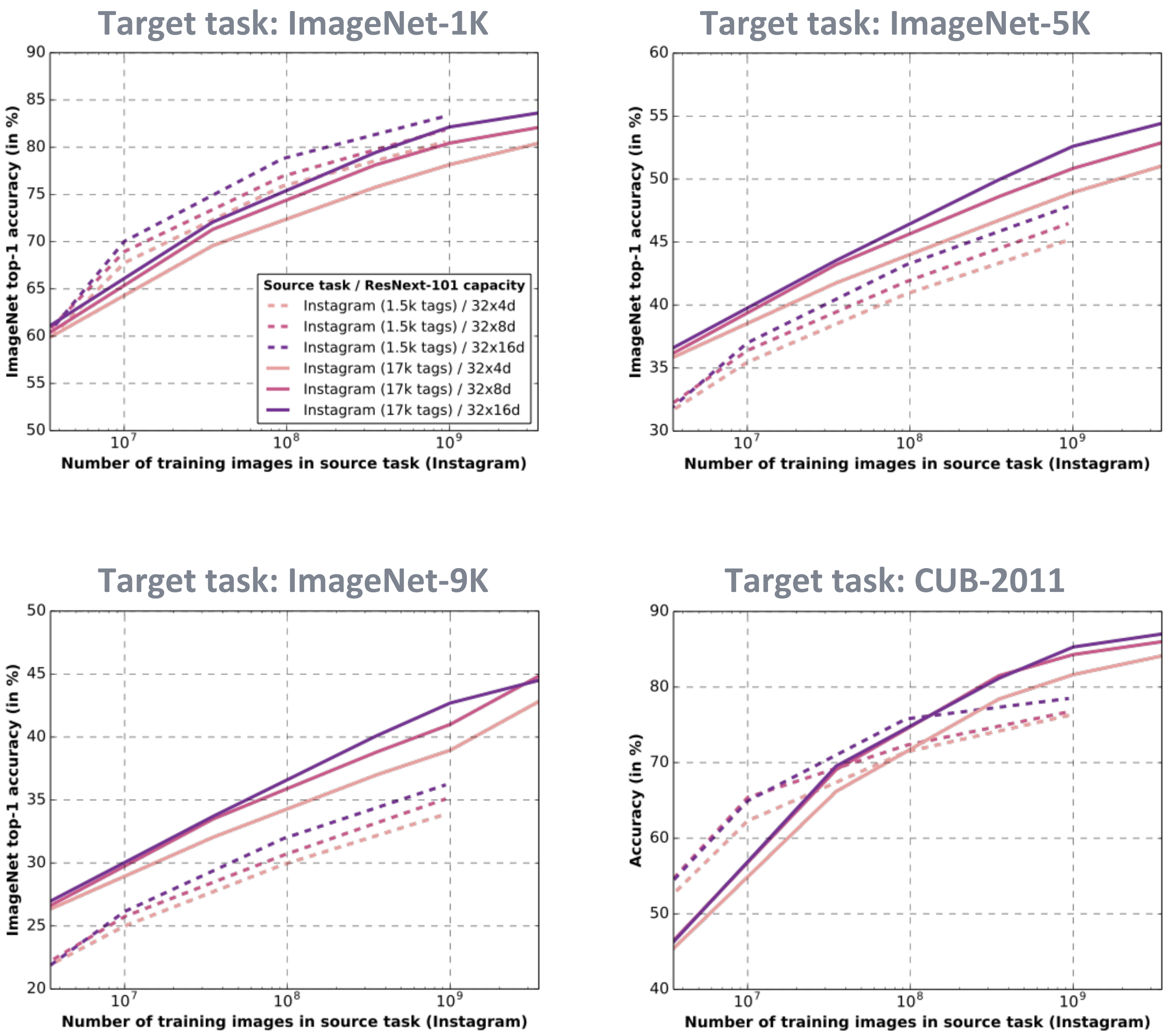
- Compared to prior SotA, +2.7% in top-1 accuracy

  (+1.4% top-5 accuracy)

Model	Image size	Parameters	Mult-adds	Top-1 Acc. (%)	Top-5 Acc. (%)
Inception V2 [27]	224	11.2M	1.94B	74.8	92.2
NASNet-A (5 @ 1538) [31]	299	10.9M	2.35B	78.6	94.2
Inception V3 [51]	299	23.8M	5.72B	78.0	93.9
Xception [52]	299	22.8M	8.38B	79.0	94.5
Inception ResNet V2 [53]	299	55.8M	13.2B	80.4	95.3
NASNet-A (7 @ 1920) [31]	299	22.6M	4.93B	80.8	95.3
ResNeXt-101 64×4 [15]	320	83.6M	31.5B	80.9	95.6
PolyNet [54]	331	92M	34.7B	81.3	95.8
DPN-131 [55]	320	79.5M	32.0B	81.5	95.8
SENet [56]	320	145.8M	42.3B	82.7	96.2
NASNet-A (6 @ 4032) [31]	331	88.9M	23.8B	82.7	96.2
Our models:					
IG-3.5B-17k ResNeXt-101 32×16d	224	194M	36B	84.2	97.2
IG-940M-1.5k ResNeXt-101 32×32d	224	466M	87B	85.1	97.5
IG-940M-1.5k ResNeXt-101 32×48d	**224**	**829M**	**153B**	**85.4**	**97.6**

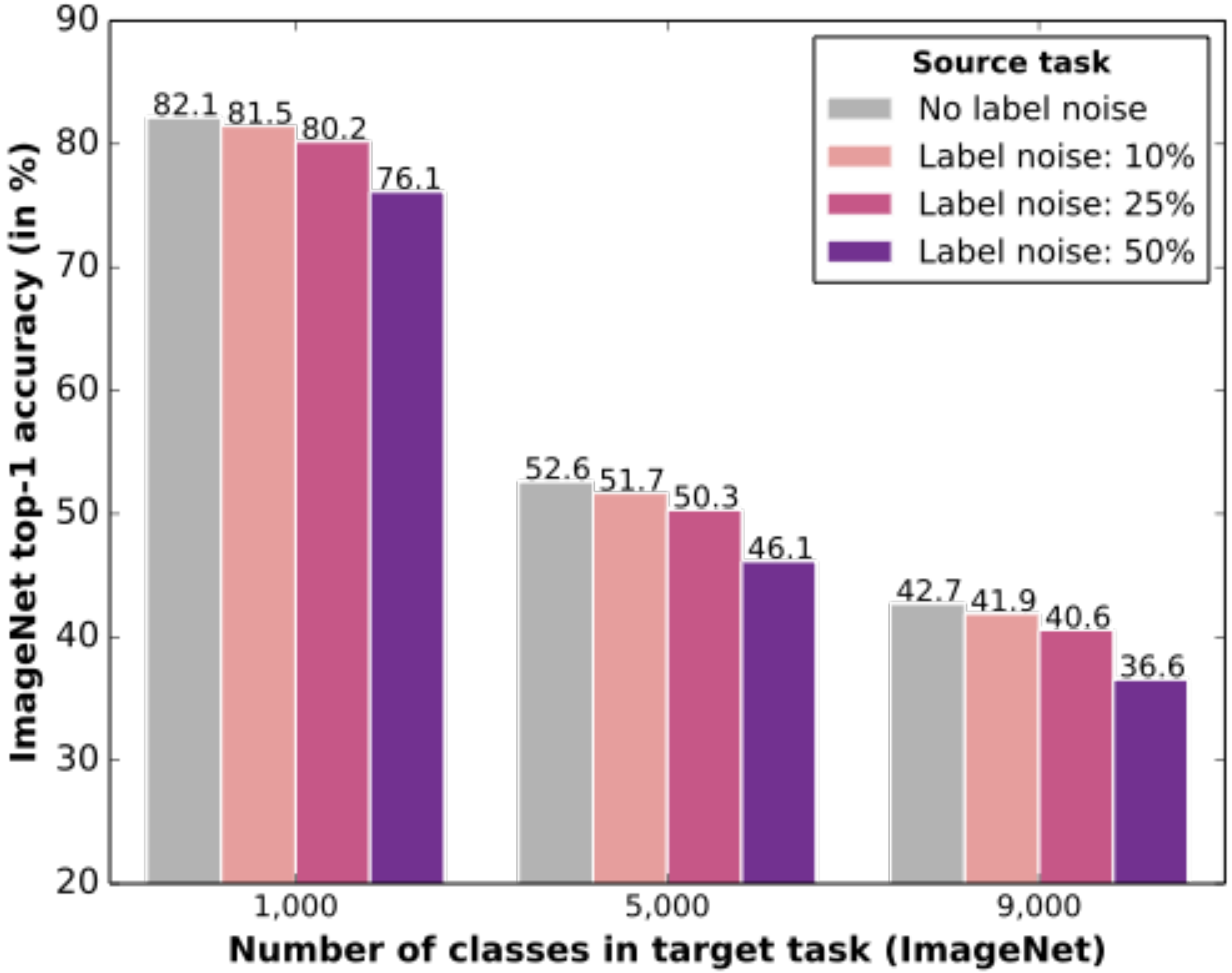
Learning Curves

- Accuracy on target task improves (almost) log-linearly with data size
- Matching hashtags to target task helps (1.5K tags)
- Positive effect of pre-training increases with difficulty of target task



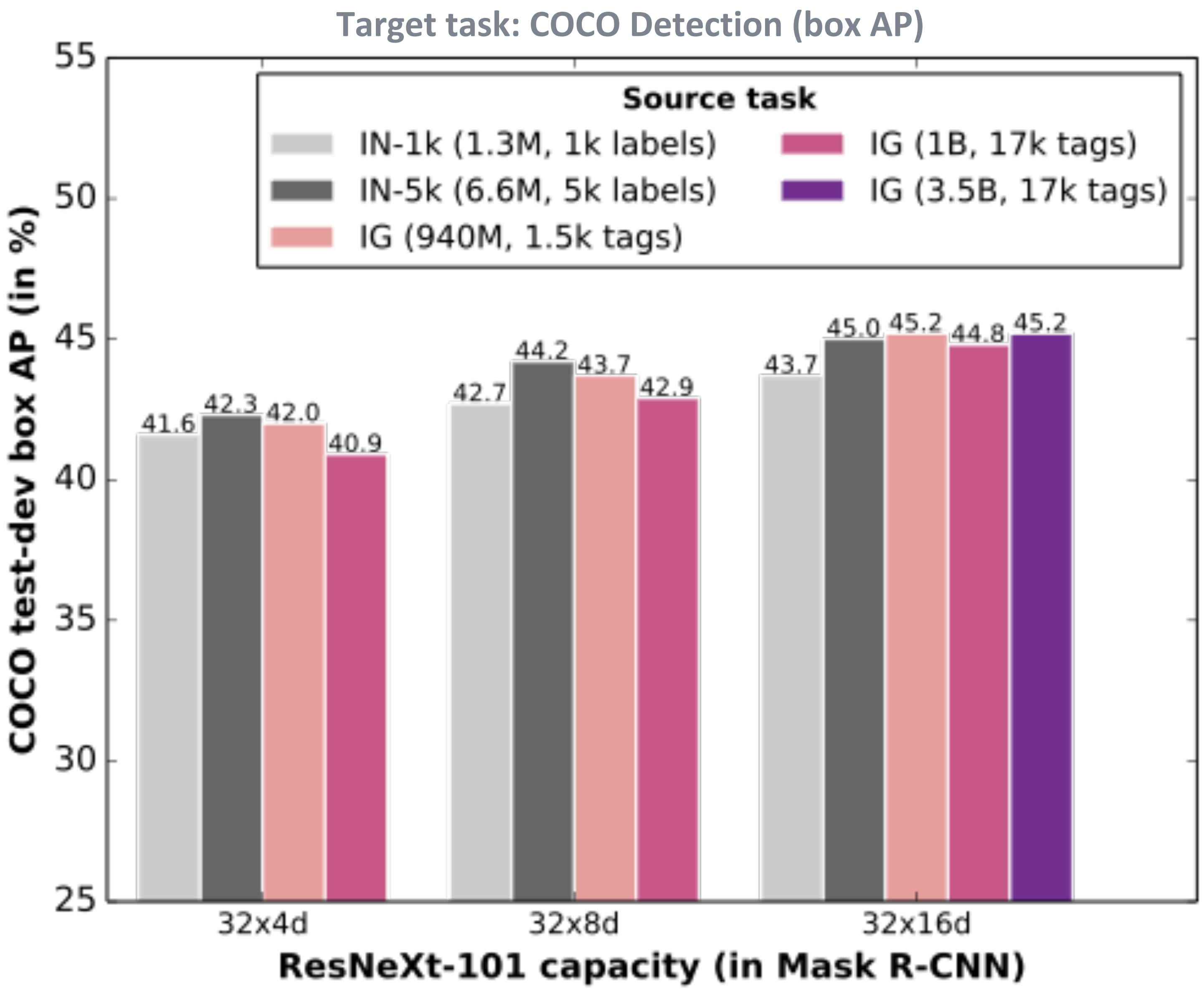
Label Noise

- Add "noise" that changes a hashtag with probability *p*
- Models are surprisingly robust to label "noise" in source task



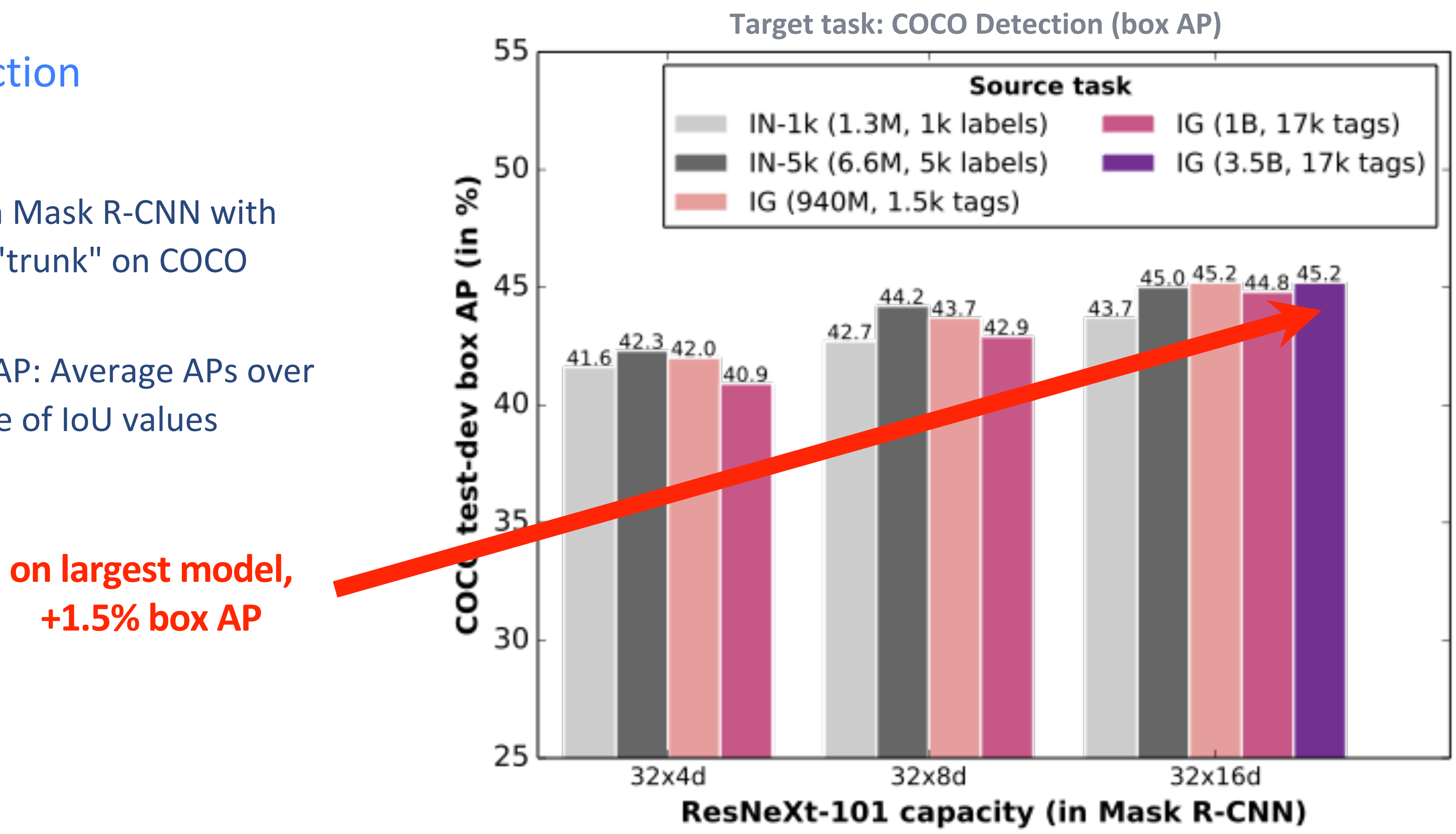
Detection

- Train Mask R-CNN with Uru "trunk" on COCO
- Box AP: Average APs over range of IoU values



Detection

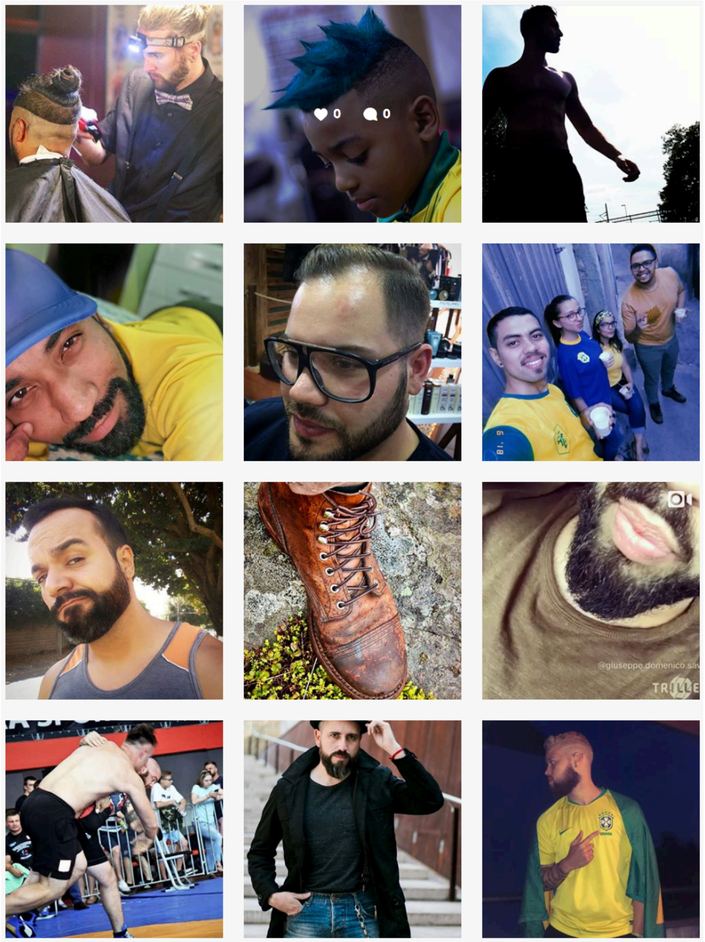
- Train Mask R-CNN with Uru "trunk" on COCO
- Box AP: Average APs over range of IoU values



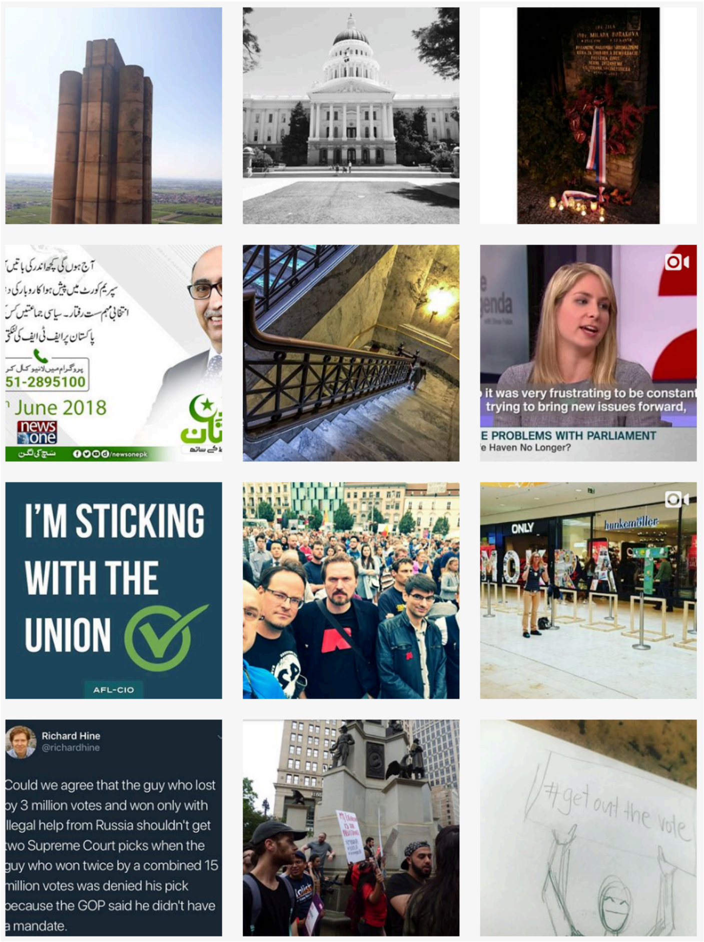
Visual Concreteness

- Predicting hashtags is easier for visually "concrete" hashtags?

#beard: concreteness = 4.96



#democracy: concreteness = 1.78

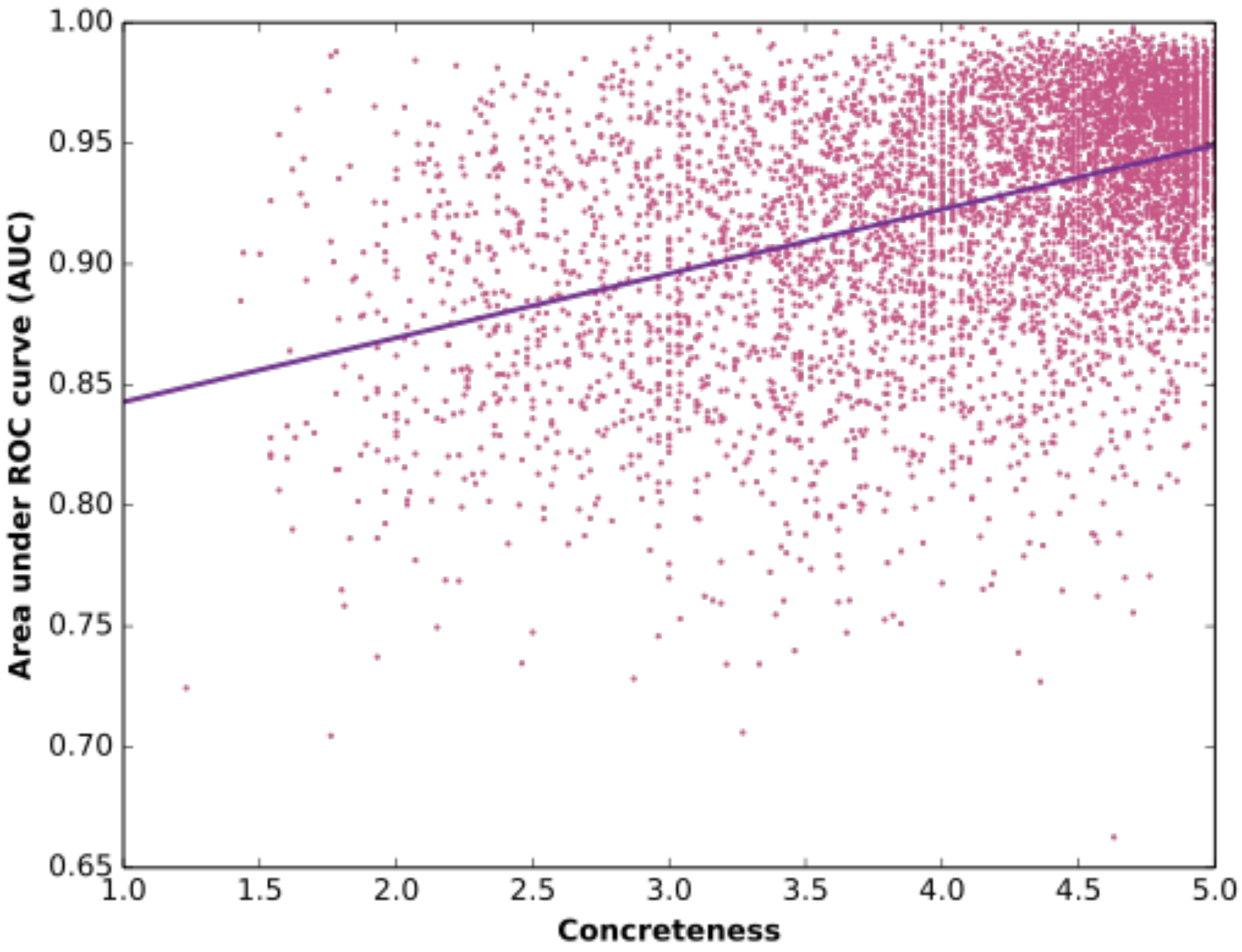


* Brysbaert *et al.*, 2014

Visual Concreteness

- Predicting hashtags is easier for visually "concrete" hashtags

- Correlation: $\rho = 0.43$



* Brysbaert *et al.*, 2014

Conclusion

Billion-scale pretraining leads to >2.0% reduction in ImageNet top-1 error

Discussion

- Results suggest further improvements are possible
- Current networks are underfitting on datasets at this scale
- Hypothesis: hashtag-based pre-training particularly beneficial as target task involves recognition of larger visual variety