Challenges and New Paradigms

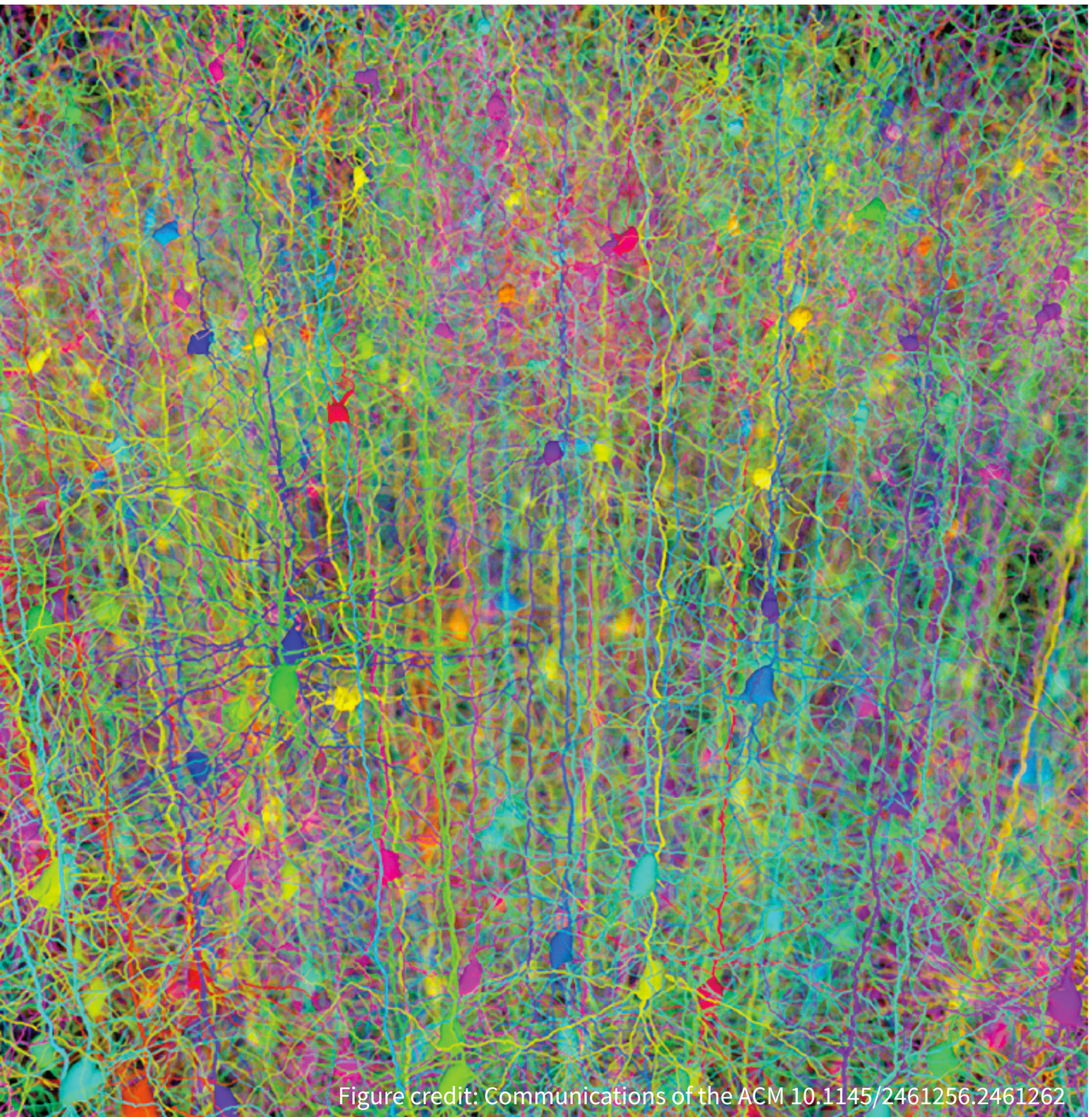


Figure credit: Communications of the ACM 10.1145/2461256.2461262

GRAHAM TAYLOR

VECTOR INSTITUTE

SCHOOL OF ENGINEERING
UNIVERSITY OF GUELPH

CANADIAN INSTITUTE
FOR ADVANCED RESEARCH

UNIVERSITY of GUELPH
CHANGING LIVES
IMPROVING LIFE

CIFAR
CANADIAN
INSTITUTE
FOR
ADVANCED
RESEARCH

Practical Issue: Normalization

- It is often useful to normalize the data so that it is consistent in some way
- Two basic forms of normalization
  - Feature normalization
  - Instance (example) normalization

Feature Normalization

Centering: $x_{n,d} \leftarrow x_{n,d} - \mu_d$

Variance Scaling: $x_{n,d} \leftarrow x_{n,d} / \sigma_d$

Absolute Scaling: $x_{n,d} \leftarrow x_{n,d} / r_d$

where:

$$\mu_d = \frac{1}{N} \sum_n x_{n,d}$$

$$\sigma_d = \sqrt{\frac{1}{N-1} \sum_n \left(x_{n,d} - \mu_d\right)^2}$$

$$r_d = \max_n \left|x_{n,d}\right|$$

Feature Normalization

Centering: $x_{n,d} \leftarrow x_{n,d} - \mu_d$

Variance Scaling: $x_{n,d} \leftarrow x_{n,d} / \sigma_d$

Absolute Scaling: $x_{n,d} \leftarrow x_{n,d} / r_d$

where:

$$\mu_d = \frac{1}{N} \sum_n x_{n,d}$$

$$\sigma_d = \sqrt{\frac{1}{N-1} \sum_n \left(x_{n,d} - \mu_d\right)^2}$$

$$r_d = \max_n \left|x_{n,d}\right|$$

What about binary data?

Example Normalization

- Most standard normalization is to ensure that the length of each example vector is one

  - Each example lies somewhere on the unit hypersphere

Example Normalization: $x_n \leftarrow x_n / ||x_n||$

DL Normalization Advice

Some simple normalization is generally useful:

- For signal data (e.g. images or audio) start with instance-based normalization
- For structured data (e.g. customer records) centre and variance-scale real-valued attributes
  - Generally don’t touch binary data
  - Normalize counts to [0, 1]
  - Don’t hesitate to log-transform (1 + x) if scales are way off

Debugging Learning Algorithms

1	real world goal	increase revenue
2	real world mechanism	better ad display
3	learning problem	classify click-through
4	data collection	interaction w/ current system
5	collected data	query, ad, click
6	data representation	bow^2, $\pm$ click
7	select model family	decision trees, depth 20
8	select training data	subset from april'16
9	train model & hyperparams	final decision tree
10	predict on test data	subset from may'16
11	evaluate error	zero/one loss for $\pm$ click
12	deploy!	(hope we achieve our goal)

Typical design process
for a ML application

via Hal Daume III ciml.info

Debugging Learning Algorithms

- Is the problem with generalization to the test data?
  - Can you fit the training data?

1	real world goal	increase revenue
2	real world mechanism	better ad display
3	learning problem	classify click-through
4	data collection	interaction w/ current system
5	collected data	query, ad, click
6	data representation	bow^2, $\pm$ click
7	select model family	decision trees, depth 20
8	select training data	subset from april'16
9	train model & hyperparams	final decision tree
10	predict on test data	subset from may'16
11	evaluate error	zero/one loss for $\pm$ click
12	deploy!	(hope we achieve our goal)

Typical design process for a ML application

via Hal Daume III ciml.info

Debugging Learning Algorithms

- Is the problem with generalization to the test data?
  - Can you fit the training data?
- Do you have a train/test mismatch?
  - Try shuffling your train + testing data together and randomly selecting a new test set

1	real world goal	increase revenue
2	real world mechanism	better ad display
3	learning problem	classify click-through
4	data collection	interaction w/ current system
5	collected data	query, ad, click
6	data representation	bow^2, $\pm$ click
7	select model family	decision trees, depth 20
8	select training data	subset from april'16
9	train model & hyperparams	final decision tree
10	predict on test data	subset from may'16
11	evaluate error	zero/one loss for $\pm$ click
12	deploy!	(hope we achieve our goal)

Typical design process for a ML application

via Hal Daume III ciml.info

Debugging Learning Algorithms

- Is the problem with generalization to the test data?
  - Can you fit the training data?
- Do you have a train/test mismatch?
  - Try shuffling your train + testing data together and randomly selecting a new test set
- Is your learning algorithm implemented correctly?
  - Instead of accuracy, try measuring whatever your model is supposedly optimizing
  - Compare against a reference implementation

1	real world goal	increase revenue
2	real world mechanism	better ad display
3	learning problem	classify click-through
4	data collection	interaction w/ current system
5	collected data	query, ad, click
6	data representation	bow^2, $\pm$ click
7	select model family	decision trees, depth 20
8	select training data	subset from april'16
9	train model & hyperparams	final decision tree
10	predict on test data	subset from may'16
11	evaluate error	zero/one loss for $\pm$ click
12	deploy!	(hope we achieve our goal)

Typical design process for a ML application

via Hal Daume III ciml.info

Debugging Learning Algorithms

- Is the problem with generalization to the test data?
  - Can you fit the training data?
- Do you have a train/test mismatch?
  - Try shuffling your train + testing data together and randomly selecting a new test set
- Is your learning algorithm implemented correctly?
  - Instead of accuracy, try measuring whatever your model is supposedly optimizing
  - Compare against a reference implementation
- Do you have an adequate representation?
  - Add features correlated with what you are trying to predict

1	real world goal	increase revenue
2	real world mechanism	better ad display
3	learning problem	classify click-through
4	data collection	interaction w/ current system
5	collected data	query, ad, click
6	data representation	bow^2, $\pm$ click
7	select model family	decision trees, depth 20
8	select training data	subset from april'16
9	train model & hyperparams	final decision tree
10	predict on test data	subset from may'16
11	evaluate error	zero/one loss for $\pm$ click
12	deploy!	(hope we achieve our goal)

Typical design process for a ML application

via Hal Daume III ciml.info

Debugging Learning Algorithms

- Is the problem with generalization to the test data?
  - Can you fit the training data?
- Do you have a train/test mismatch?
  - Try shuffling your train + testing data together and randomly selecting a new test set
- Is your learning algorithm implemented correctly?
  - Instead of accuracy, try measuring whatever your model is supposedly optimizing
  - Compare against a reference implementation
- Do you have an adequate representation?
  - Add features correlated with what you are trying to predict
- Do you have enough data?
  - Try training on 80% of your training data and see what happens

1	real world goal	increase revenue
2	real world mechanism	better ad display
3	learning problem	classify click-through
4	data collection	interaction w/ current system
5	collected data	query, ad, click
6	data representation	bow^2, $\pm$ click
7	select model family	decision trees, depth 20
8	select training data	subset from april'16
9	train model & hyperparams	final decision tree
10	predict on test data	subset from may'16
11	evaluate error	zero/one loss for $\pm$ click
12	deploy!	(hope we achieve our goal)

Typical design process for a ML application

via Hal Daume III ciml.info

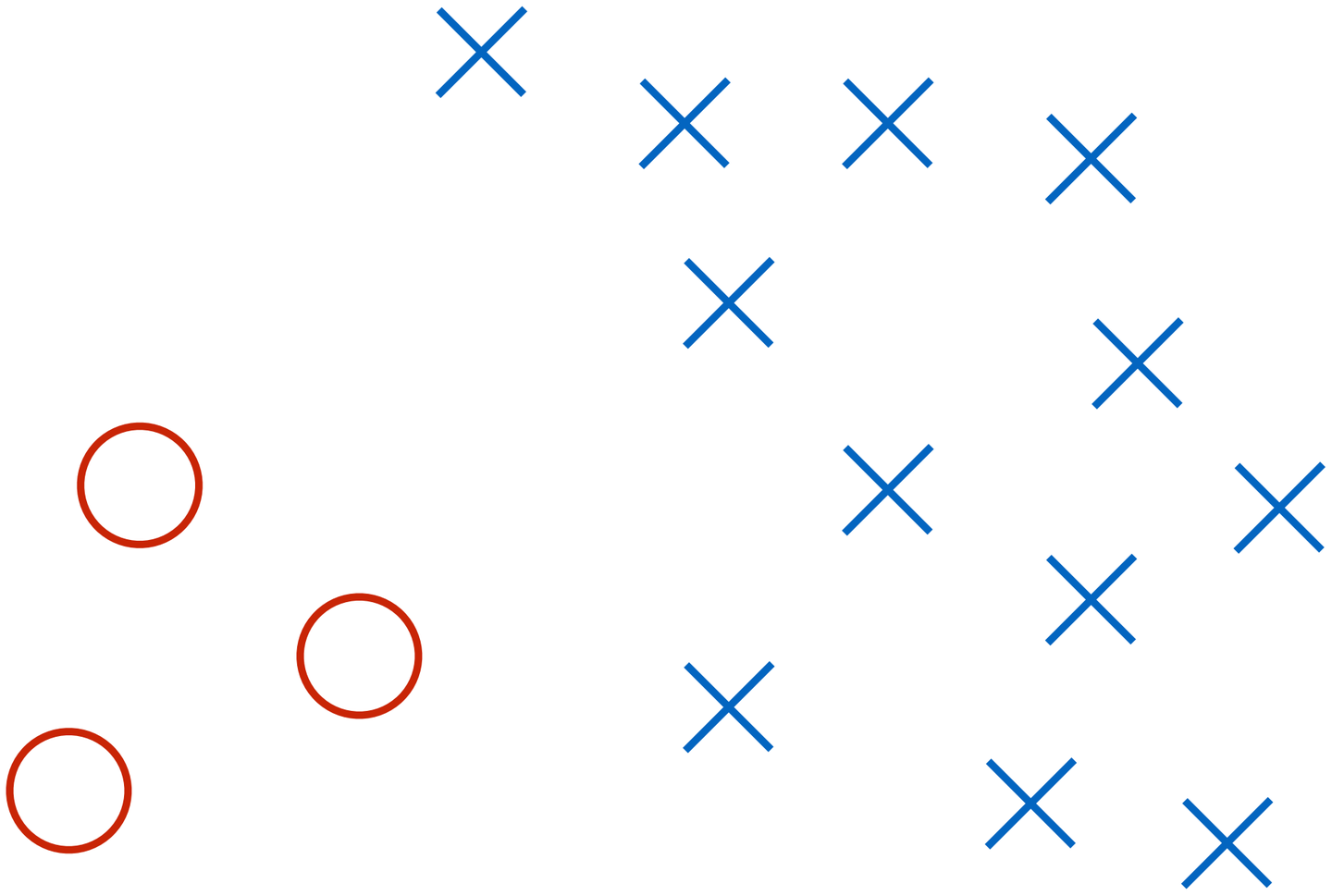
Learning with Imbalanced Data

It’s not your data that’s imbalanced, it’s the distribution you’re drawing from

- Some strategies:
- Subsampling throws away of your overrepresented data (makes learning efficient)
- Weighting, instead of throwing away overrepresented data, makes them less important
- For low-dimensional data, can try Synthetic Minority Oversampling Technique (SMOTE)

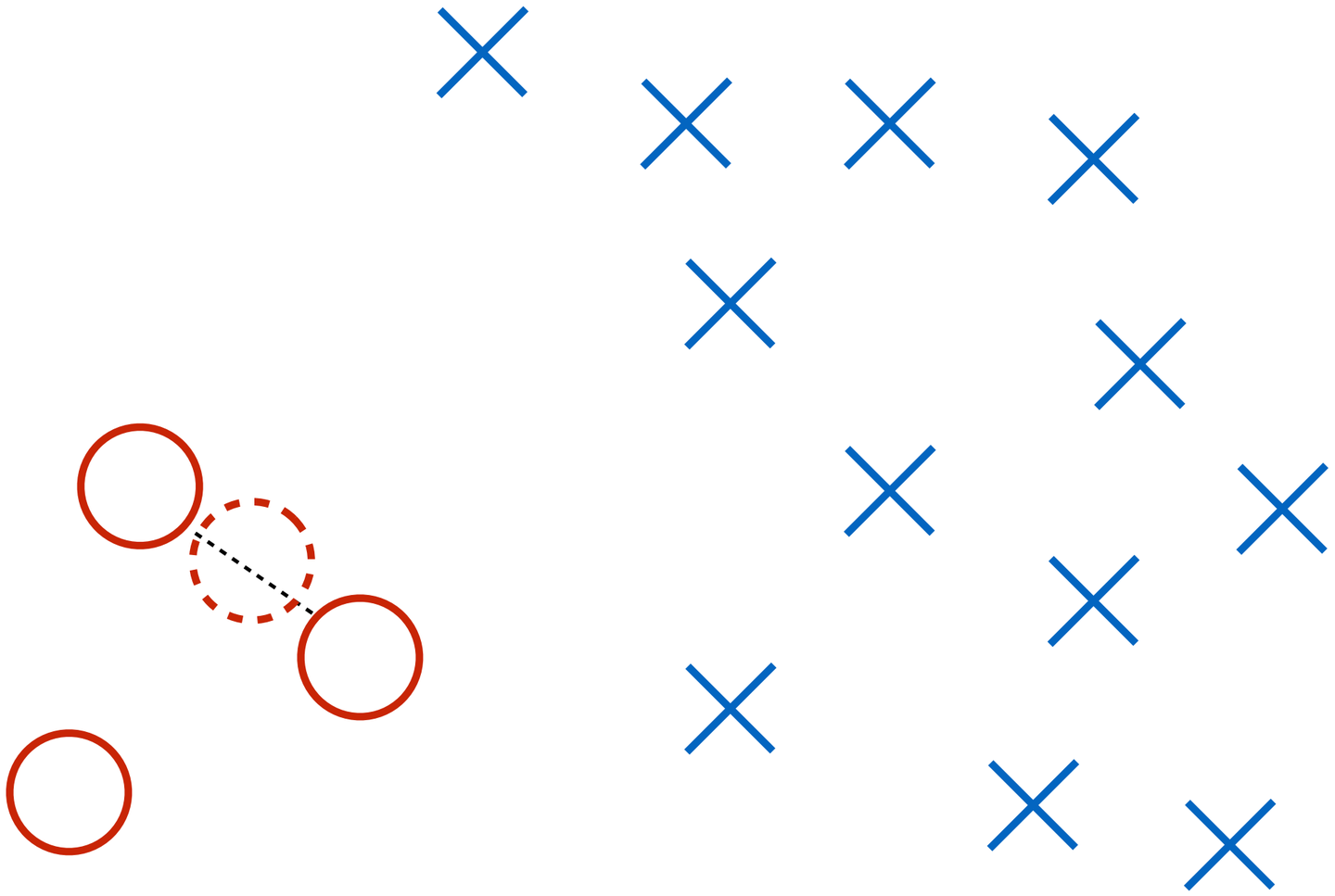
SMOTE

(Chawla et al. 2002)



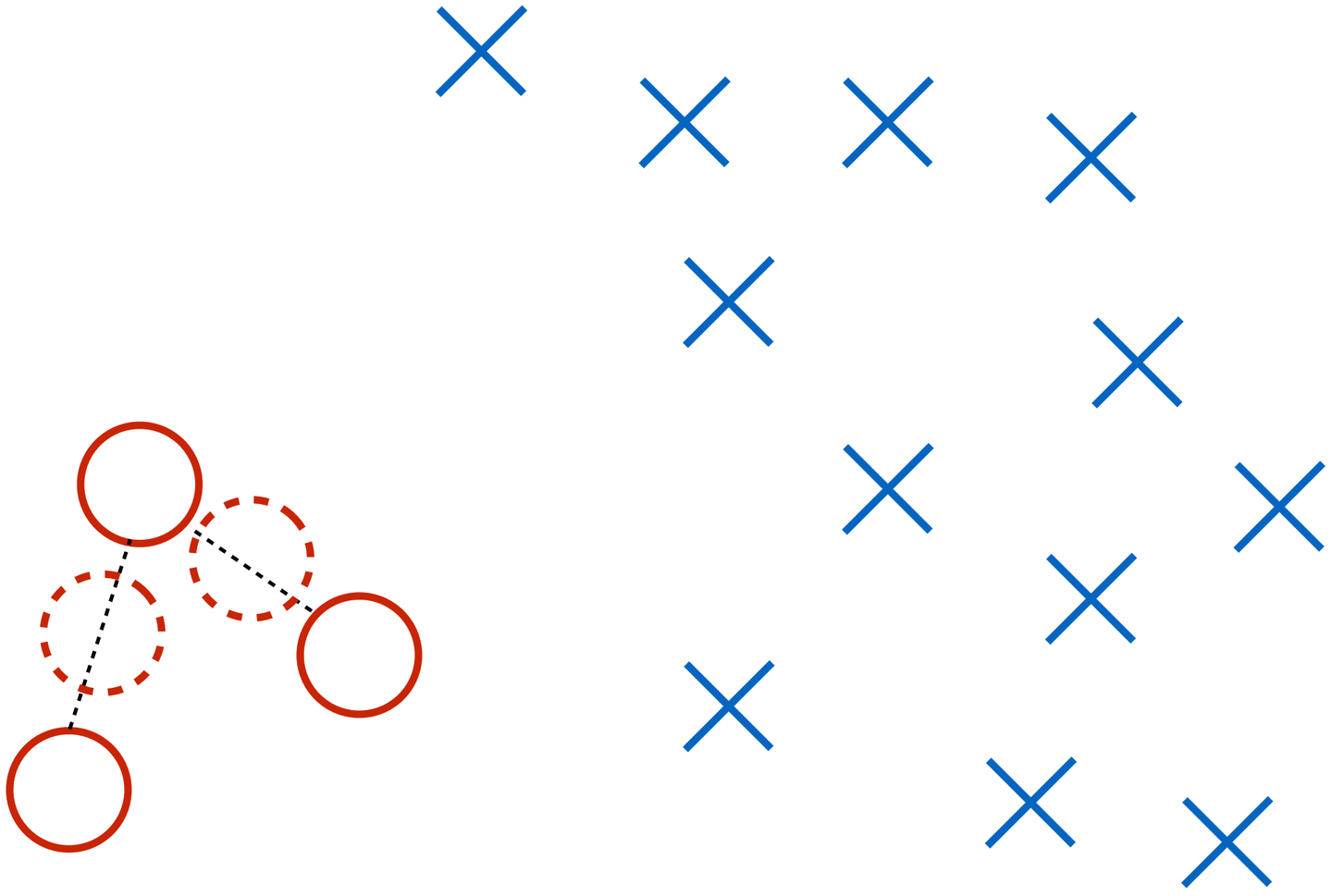
SMOTE

(Chawla et al. 2002)



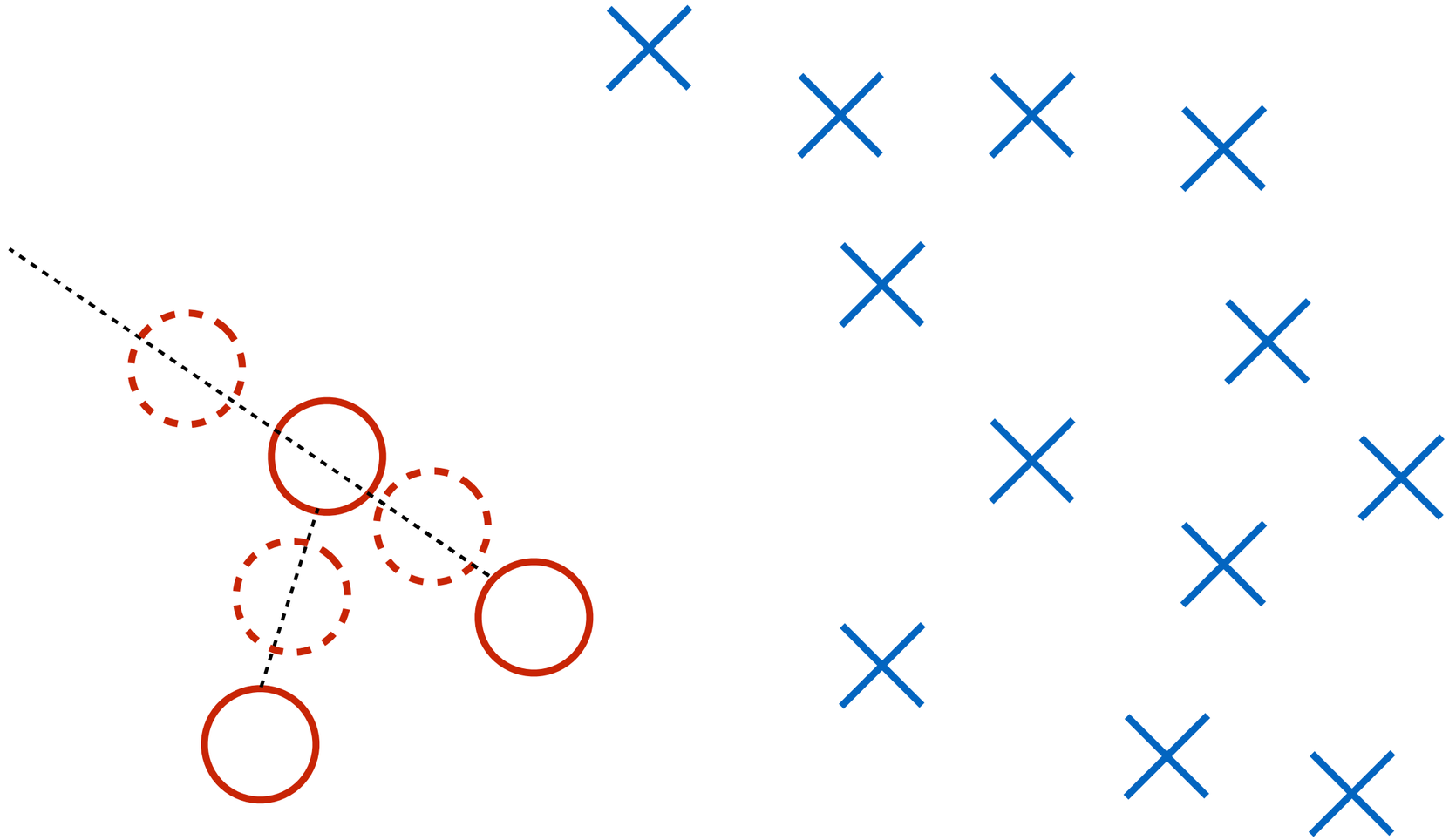
SMOTE

(Chawla et al. 2002)

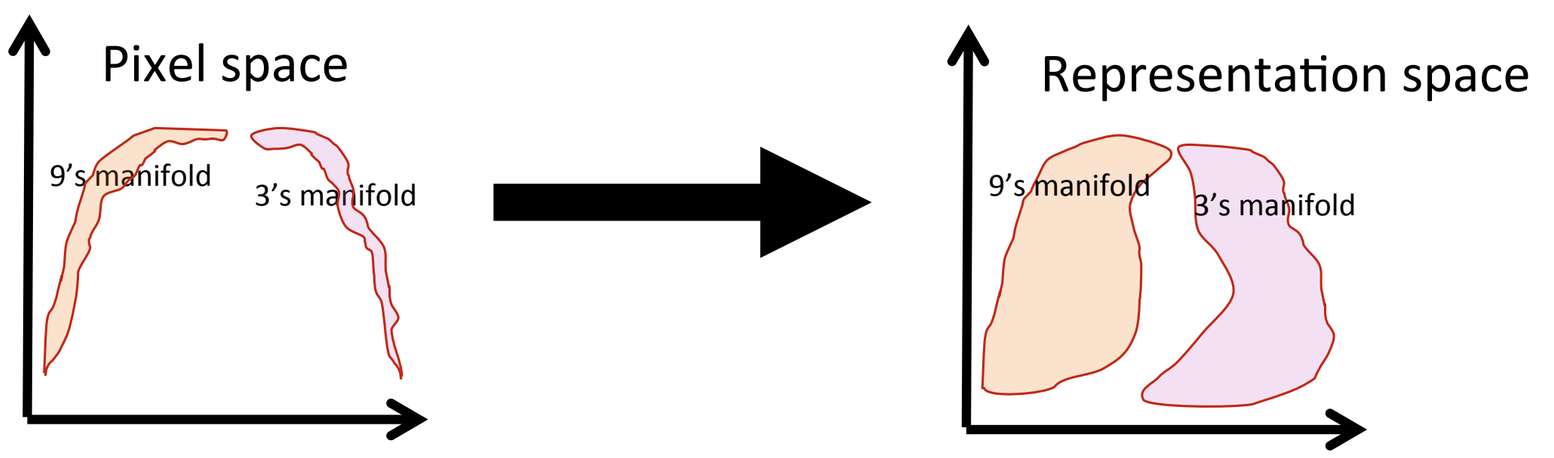


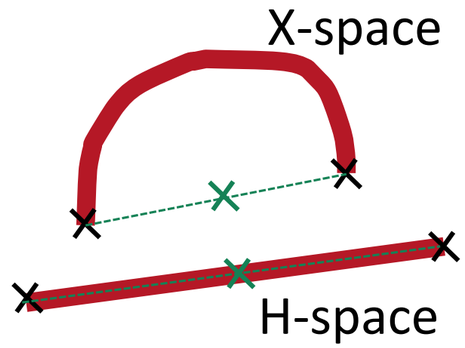
SMOTE

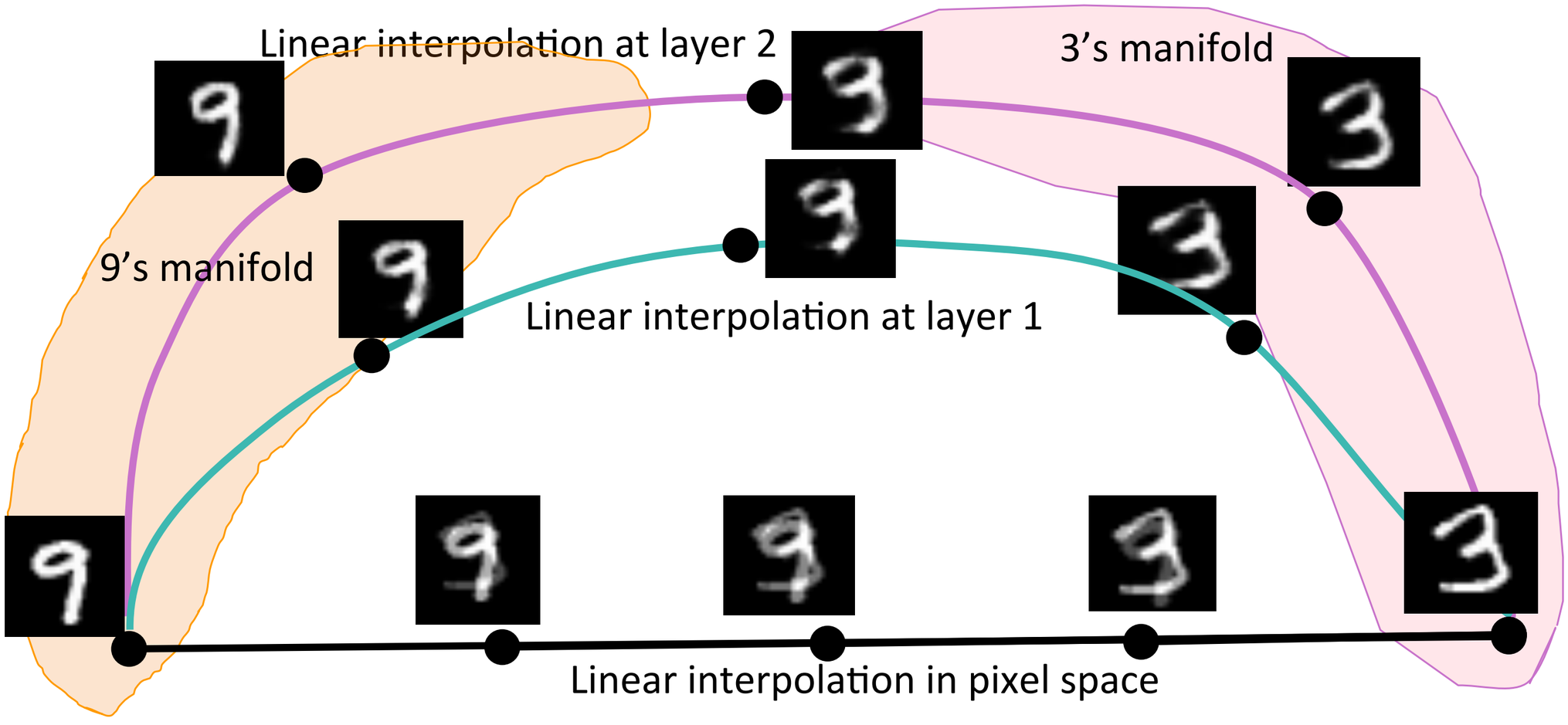
(Chawla et al. 2002)



Problem with High-D Data



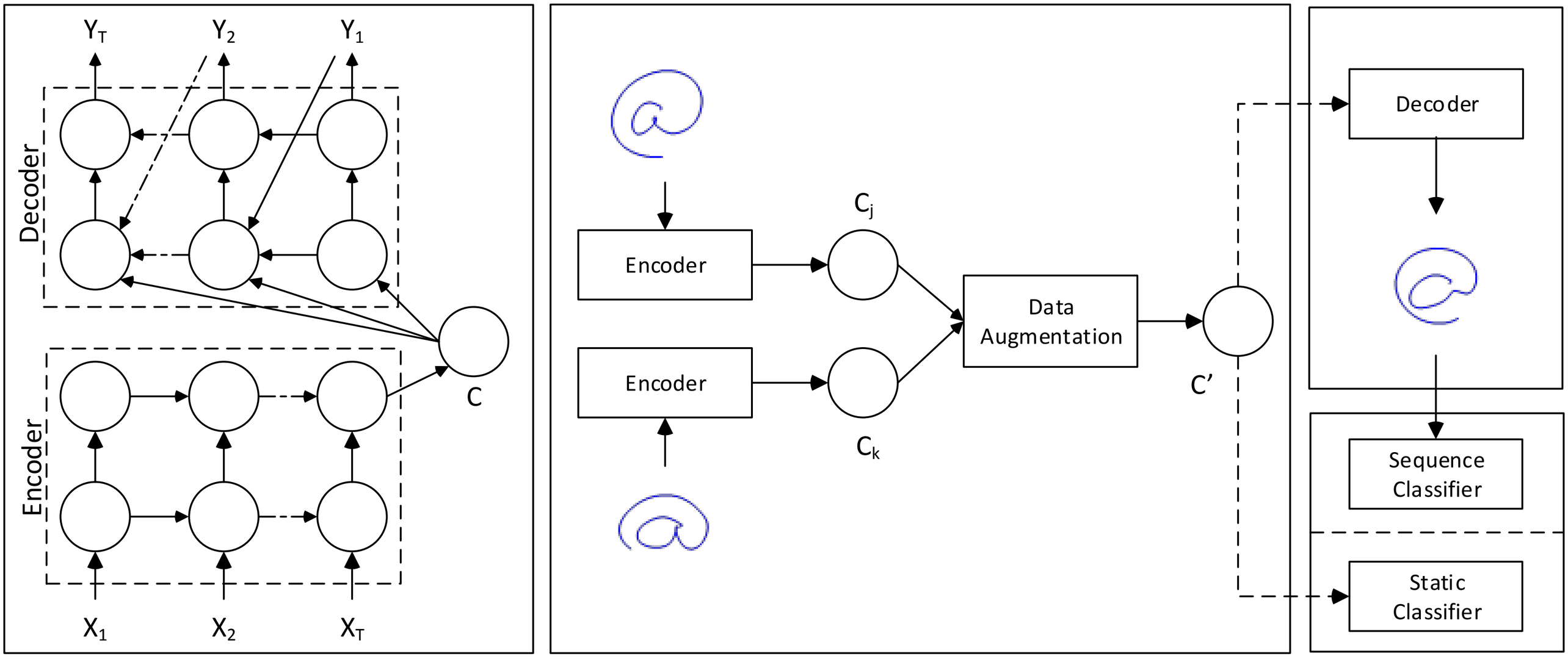




Ozair and Bengio (2014) based on Bengio et al. (2013)

(Devries and Taylor 2017)

Augmentation in Feature Space



1 Train seq2seq AE (unsupervised)

2 Data is encoded to context vectors (C) and then augmented

3 The resulting C’s are either classified directly using a static classifier, or decoded to sequences for training a sequence classifier

Via Opel

Via SurveyMonkey

Adaptation Challenge

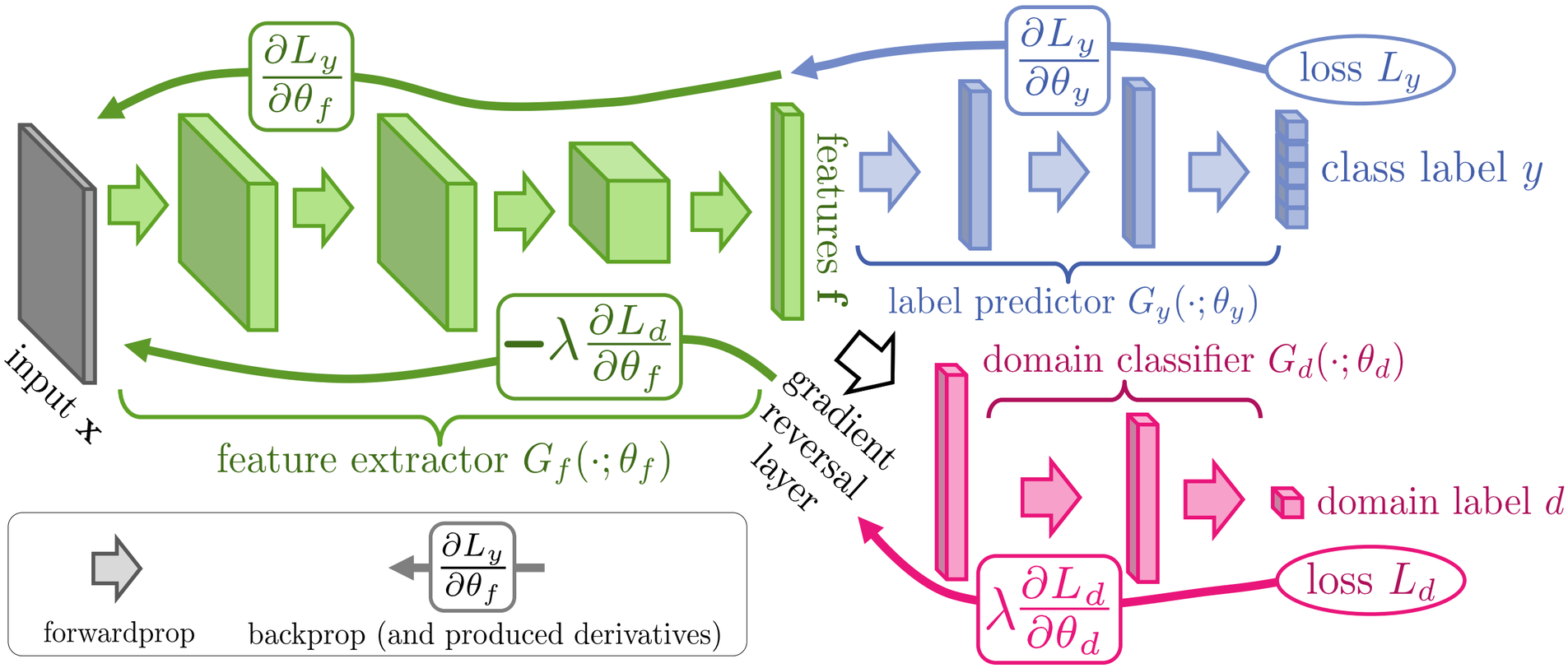
The issue of train/test mismatch has been studied under many different names

- Covariate shift
- Sample selection bias
- Domain adaptation

The adaptation challenge is: given training data from one "old" distribution, learn a classifier that does a good job on another related but different "new" distribution

via Hal Daume III ciml.info

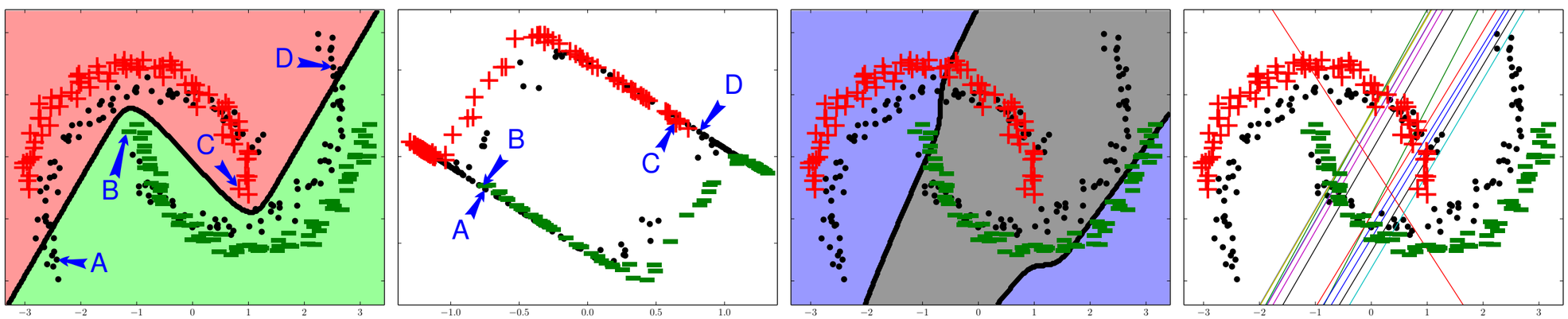
Domain Adversarial/ Gradient Reversal Layer



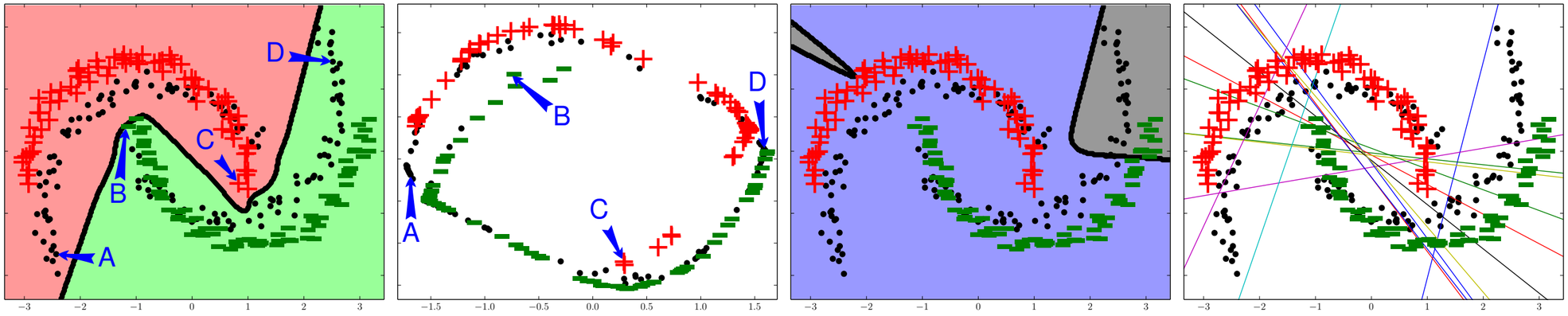
Ganin et al. (2016)

Understanding DANN

Label Classification | PCA on Representation | Domain Classification | Hidden Representation



Above. NN trained with domain classifier, no GRL



Above. Same NN architecture but with GRL - Domain Adversarial Neural Network

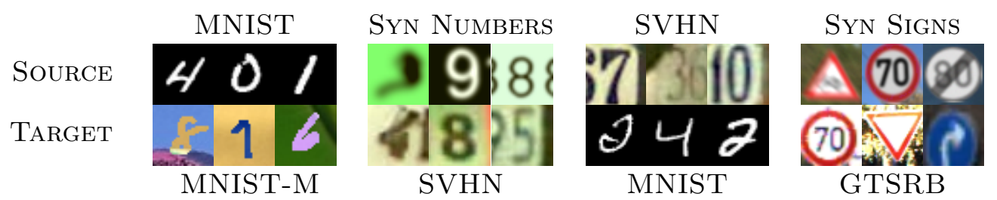
Ganin et al. (2016)

Experiments on Real Data

Amazon reviews

		Original data			mSDA representation		
Source	Target	DANN	NN	SVM	DANN	NN	SVM
books	dvd	.784	.790	**.799**	.829	.824	**.830**
books	electronics	.733	.747	**.748**	**.804**	.770	.766
books	kitchen	**.779**	.778	.769	**.843**	.842	.821
dvd	books	.723	.720	**.743**	.825	.823	**.826**
dvd	electronics	**.754**	.732	.748	**.809**	.768	.739
dvd	kitchen	**.783**	.778	.746	.849	**.853**	.842
electronics	books	**.713**	.709	.705	**.774**	.770	.762
electronics	dvd	**.738**	.733	.726	**.781**	.759	.770
electronics	kitchen	**.854**	**.854**	.847	.881	**.863**	.847
kitchen	books	**.709**	.708	.707	.718	.721	**.769**
kitchen	dvd	**.740**	.739	.736	**.789**	**.789**	.788
kitchen	electronics	**.843**	.841	.842	.856	.850	**.861**

Image Classification



Method	Source	MNIST	Syn Numbers	SVHN	Syn Signs
	Target	MNIST-M	SVHN	MNIST	GTSRB
Source only		.5225	.8674	.5490	.7900
SA (Fernando et al., 2013)		.5690 (4.1%)	.8644 (−5.5%)	.5932 (9.9%)	.8165 (12.7%)
DANN		**.7666** (52.9%)	**.9109** (79.7%)	**.7385** (42.6%)	**.8865** (46.4%)
Train on target		.9596	.9220	.9942	.9980

Office dataset

Method	Source	Amazon	DSLR	Webcam
	Target	Webcam	Webcam	DSLR
GFK(PLS, PCA) (Gong et al., 2012)		.197	.497	.6631
SA* (Fernando et al., 2013)		.450	.648	.699
DLID (Chopra et al., 2013)		.519	.782	.899
DDC (Tzeng et al., 2014)		.618	.950	.985
DAN (Long and Wang, 2015)		.685	.960	.990
Source only		.642	.961	.978
DANN		**.730**	**.964**	**.992**

Ganin et al. (2016)

Semi-Supervised Learning

- Since labeled data is usually expensive to collect, it is attractive to consider learning with labeled and unlabeled data

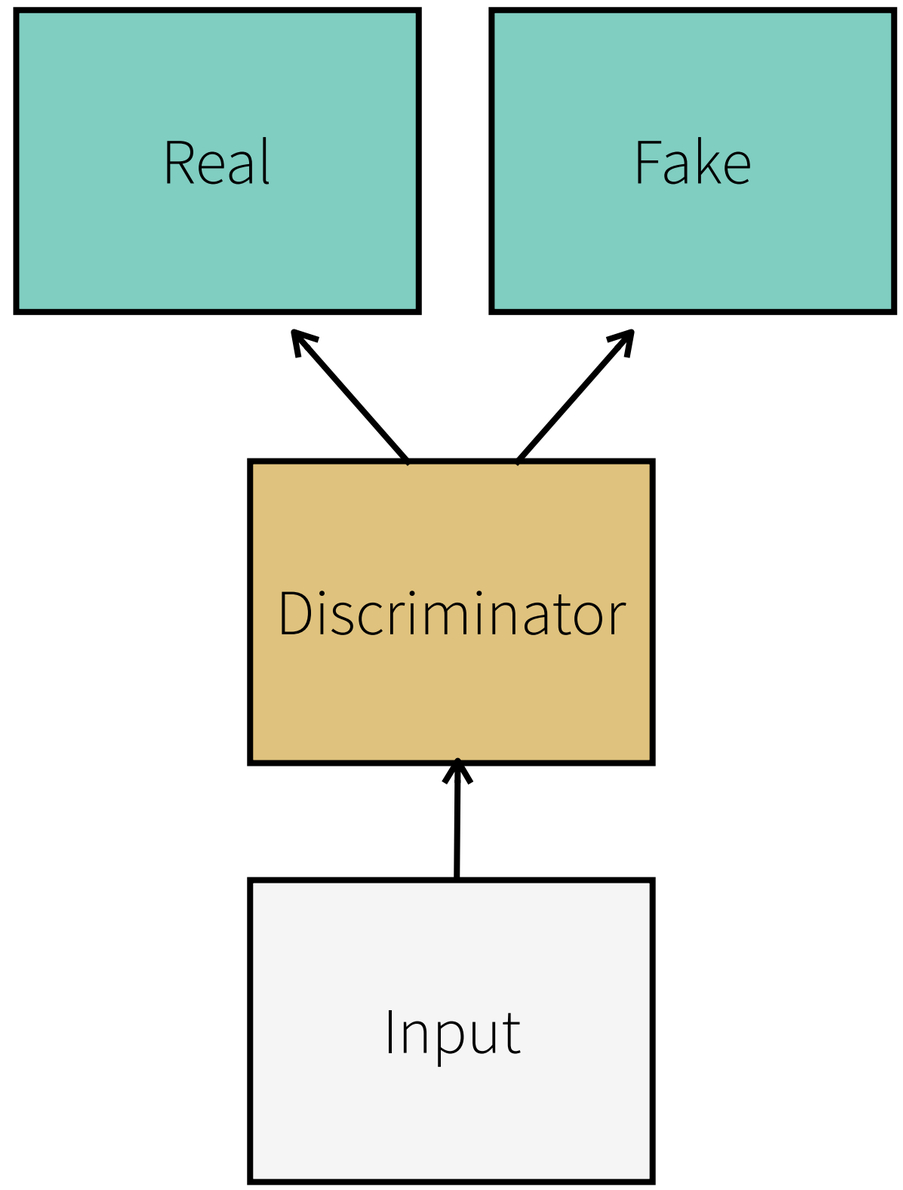
  - This is a particular case of the missing data problem

- Many of the unsupervised models we have seen so far have been applied in this setting: DBM & DBM, stacked autoencoders, GANs, VAEs, etc.

Semi-supervised GAN

Turn a classification problem with n classes into a classification problem with n + 1 classes

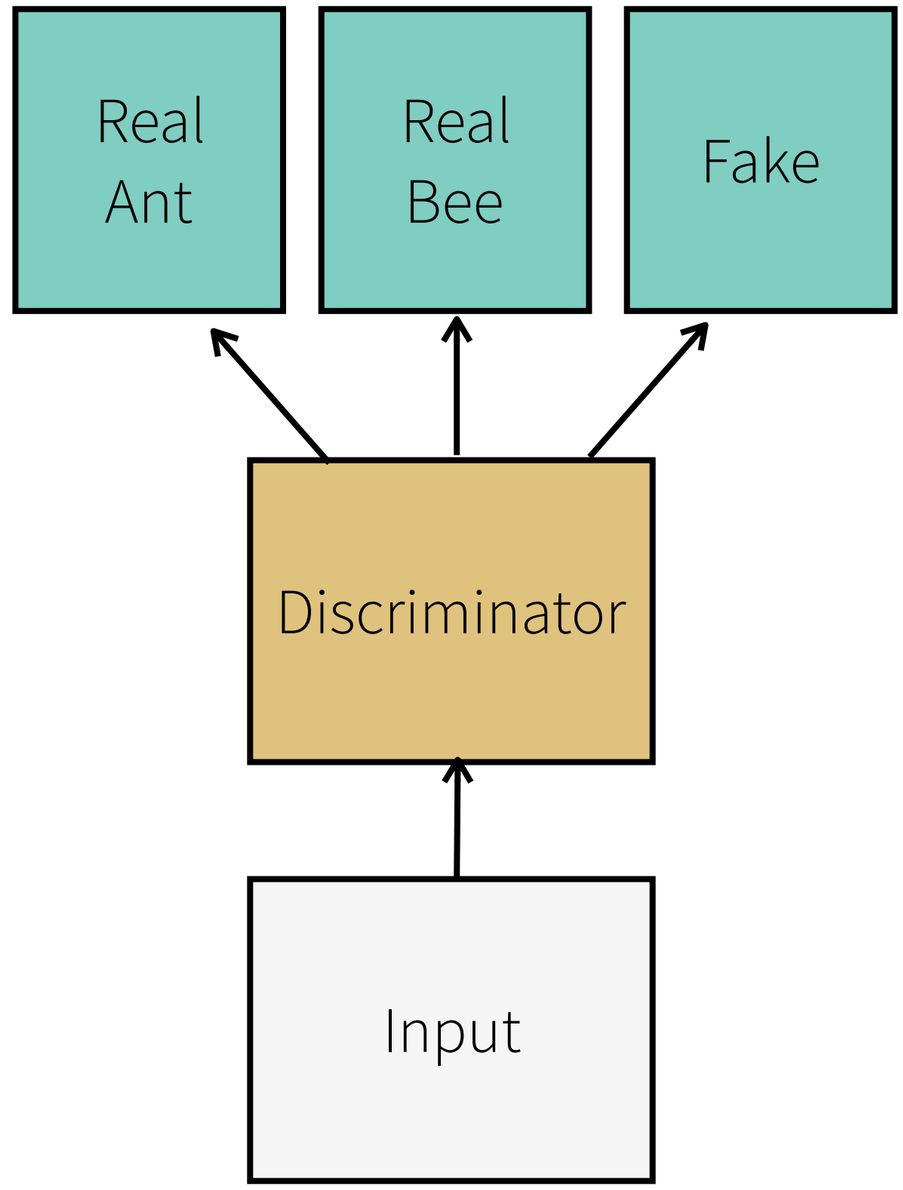
- Additional class corresponds to fake data
- The probabilities of the real classes can be summed to obtain the probability of real data



Semi-supervised GAN

Turn a classification problem with n classes into a classification problem with n + 1 classes

- Additional class corresponds to fake data
- The probabilities of the real classes can be summed to obtain the probability of real data

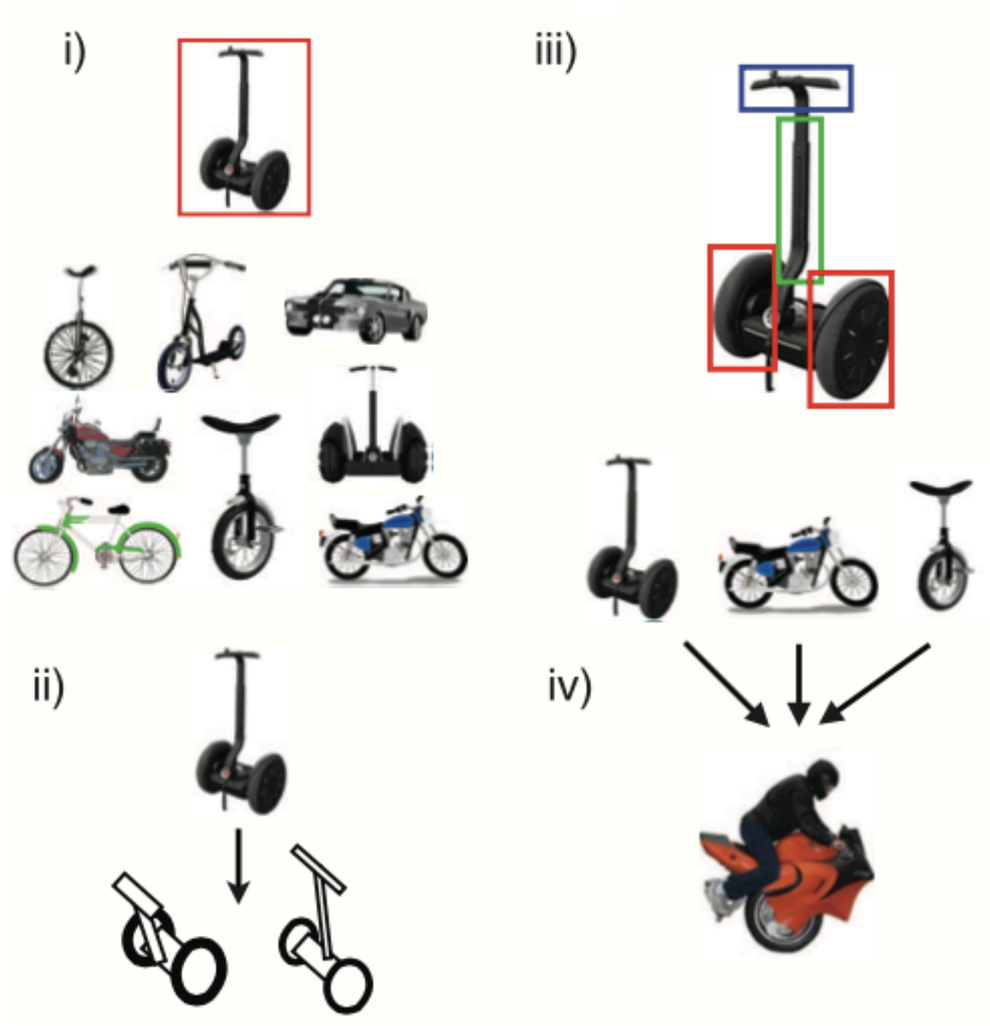


(Odena 2016, Salimans et al. 2016)

Human-level Concept Learning

Given a single example of a concept, humans can:

- Classify new examples
- Generate examples of similar type
- Parse it into parts; and
- Fantasize new examples



via Lake et al. (2015)

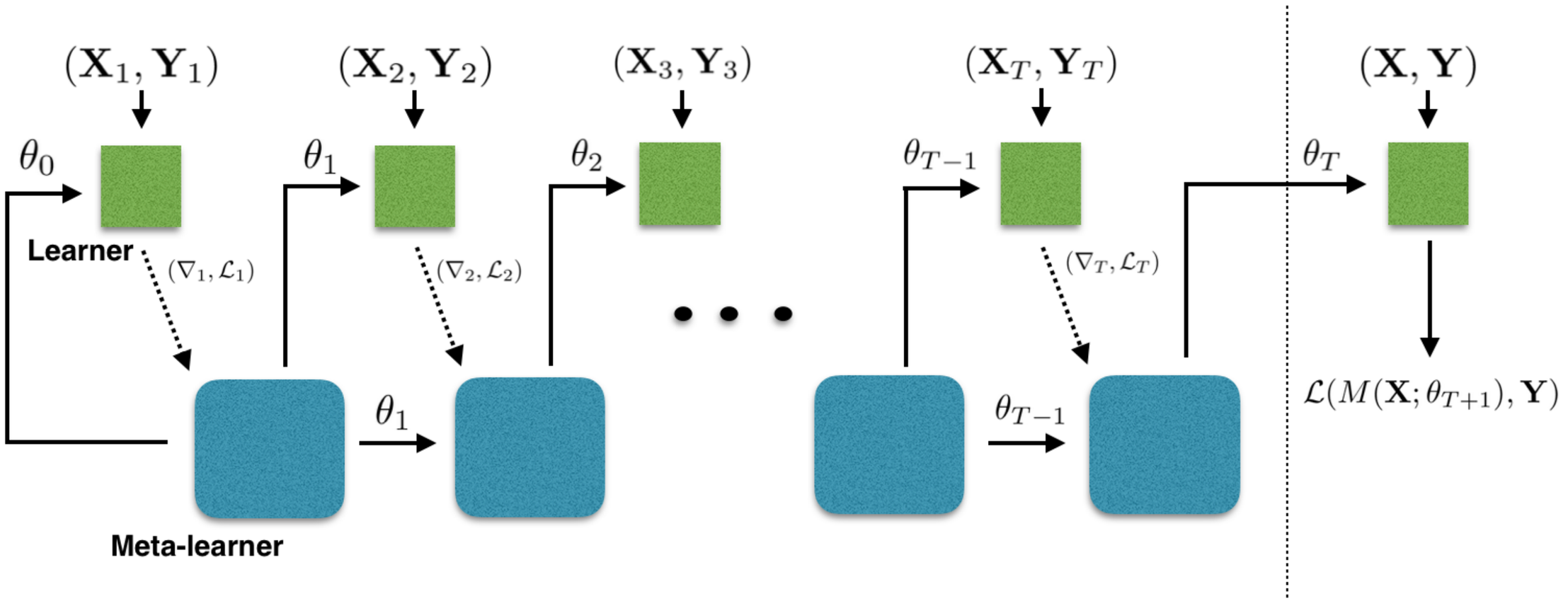
Meta-learning

These systems are trained by being exposed to a large number of tasks

Different than most machine learning setups which involve training on a single task

Adapted from Ravi and Larochelle (2016) by Finn (2017)

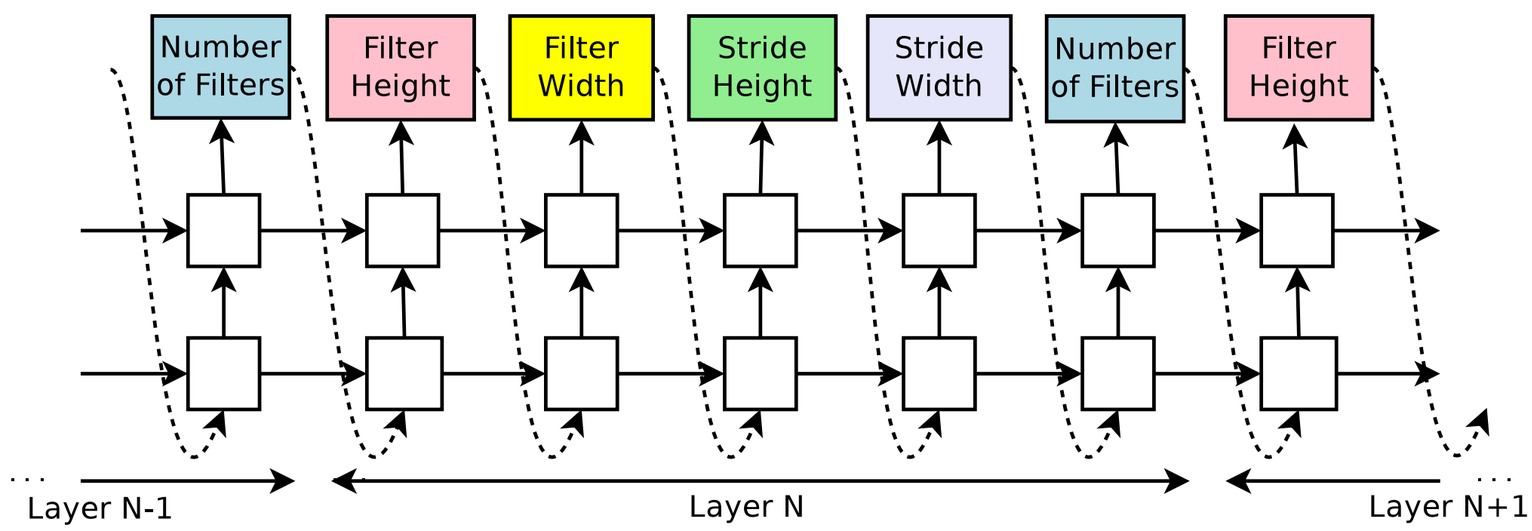
Meta-learning: Few-shot



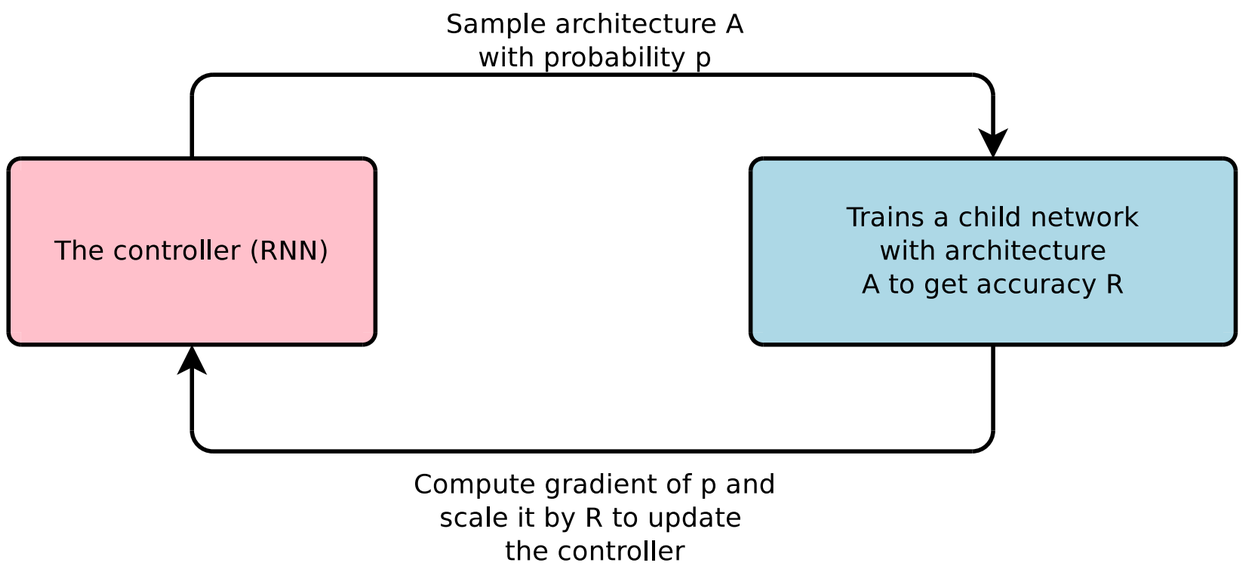
via Ravi and Larochelle (2016)

Meta-learning: Architecture

RNN controller outputs architecture as string



RNN controller trained with reinforcement learning

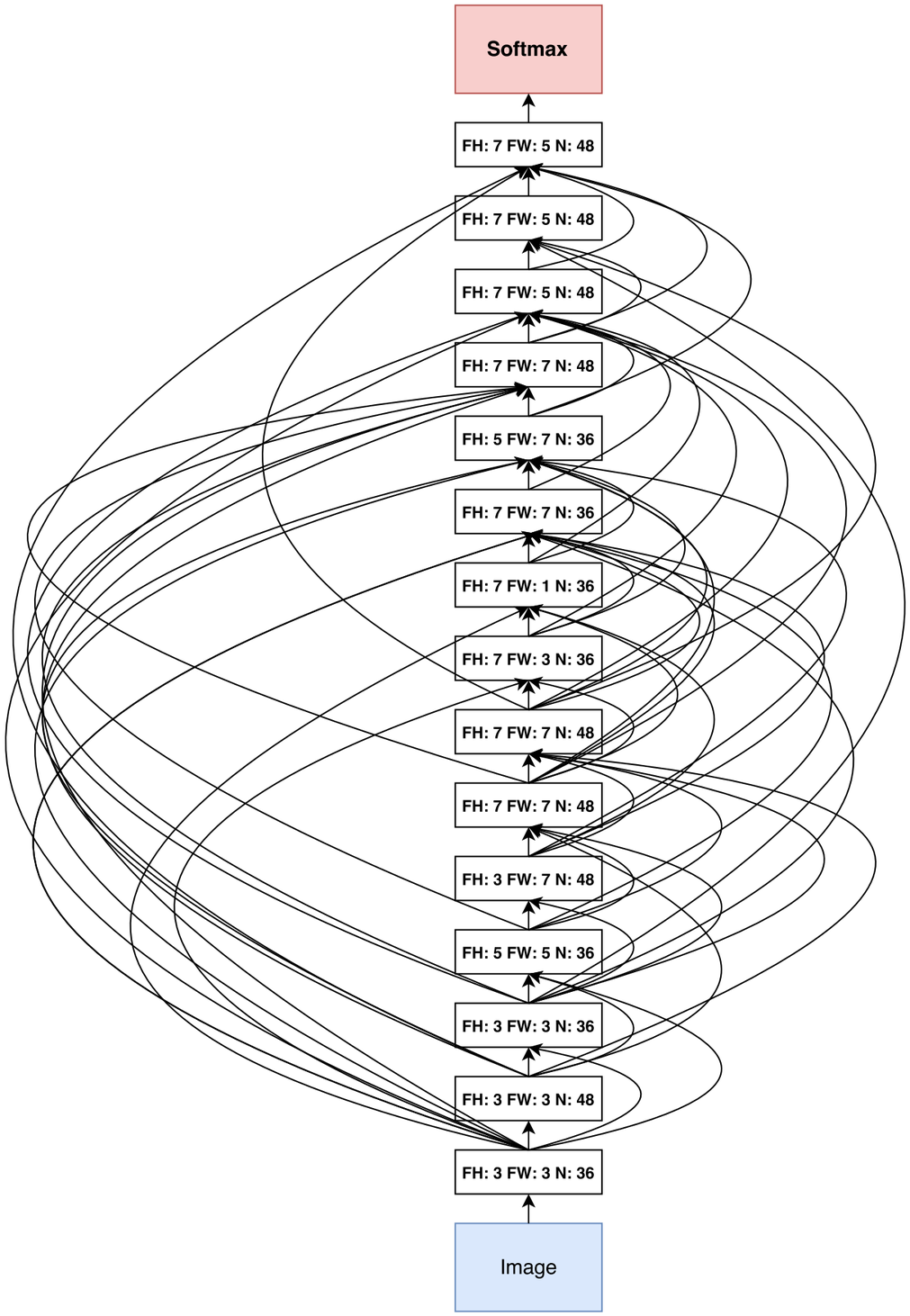


via Zoph and Le (2017)

Discovered CNN Architecture

Error rate on CIFAR

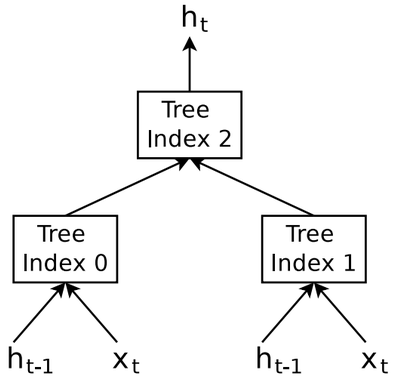
Model	Depth	Parameters	Error rate (%)
Network in Network (Lin et al., 2013)	-	-	8.81
All-CNN (Springenberg et al., 2014)	-	-	7.25
Deeply Supervised Net (Lee et al., 2015)	-	-	7.97
Highway Network (Srivastava et al., 2015)	-	-	7.72
Scalable Bayesian Optimization (Snoek et al., 2015)	-	-	6.37
FractalNet (Larsson et al., 2016)	21	38.6M	5.22
with Dropout/Drop-path	21	38.6M	4.60
ResNet (He et al., 2016a)	110	1.7M	6.61
ResNet (reported by Huang et al. (2016c))	110	1.7M	6.41
ResNet with Stochastic Depth (Huang et al., 2016c)	110	1.7M	5.23
	1202	10.2M	4.91
Wide ResNet (Zagoruyko & Komodakis, 2016)	16	11.0M	4.81
	28	36.5M	4.17
ResNet (pre-activation) (He et al., 2016b)	164	1.7M	5.46
	1001	10.2M	4.62
DenseNet ($L = 40, k = 12$) Huang et al. (2016a)	40	1.0M	5.24
DenseNet($L = 100, k = 12$) Huang et al. (2016a)	100	7.0M	4.10
DenseNet ($L = 100, k = 24$) Huang et al. (2016a)	100	27.2M	3.74
DenseNet-BC ($L = 100, k = 40$) Huang et al. (2016b)	190	25.6M	3.46
Neural Architecture Search v1 no stride or pooling	15	4.2M	5.50
Neural Architecture Search v2 predicting strides	20	2.5M	6.01
Neural Architecture Search v3 max pooling	39	7.1M	4.47
Neural Architecture Search v3 max pooling + more filters	39	37.4M	3.65

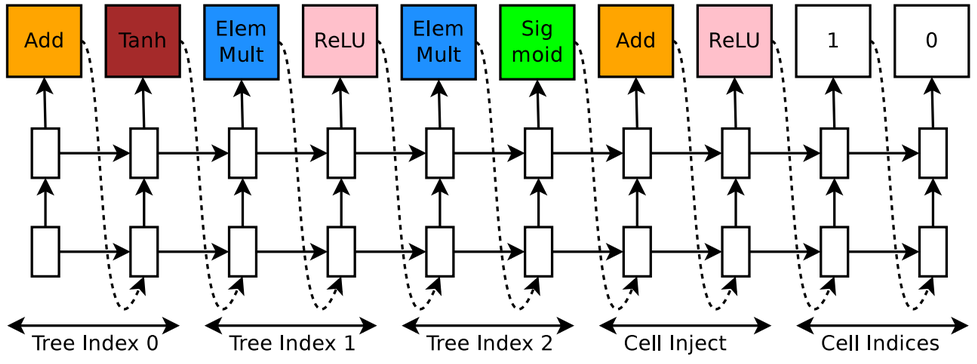


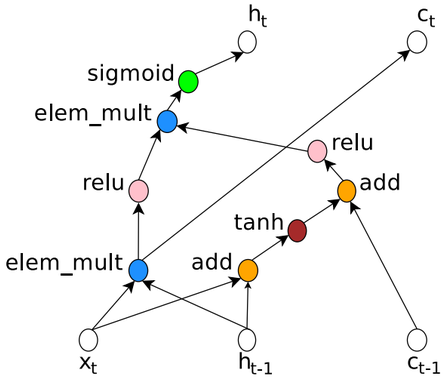
via Zoph and Le (2017)

Learns RNN Cells

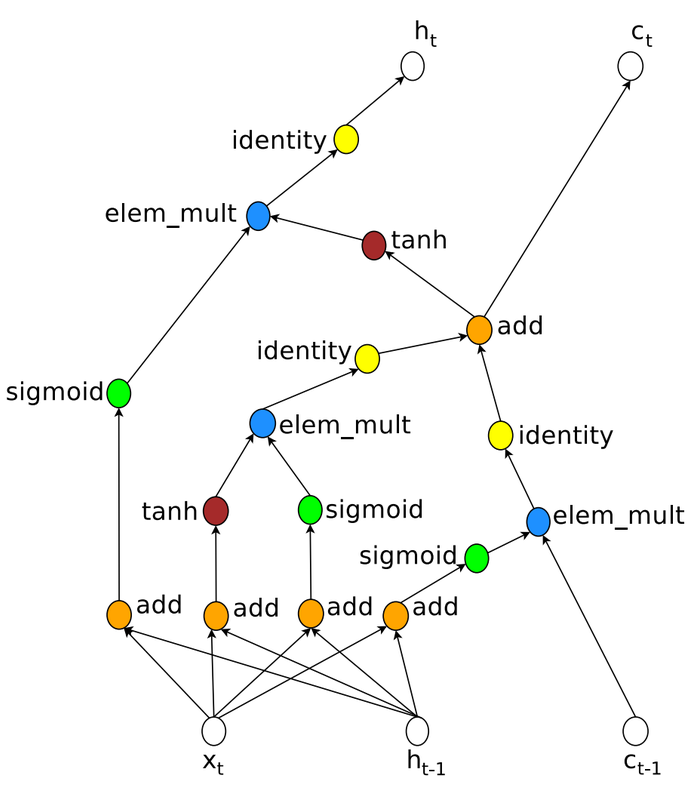
Writing RNN cells as strings



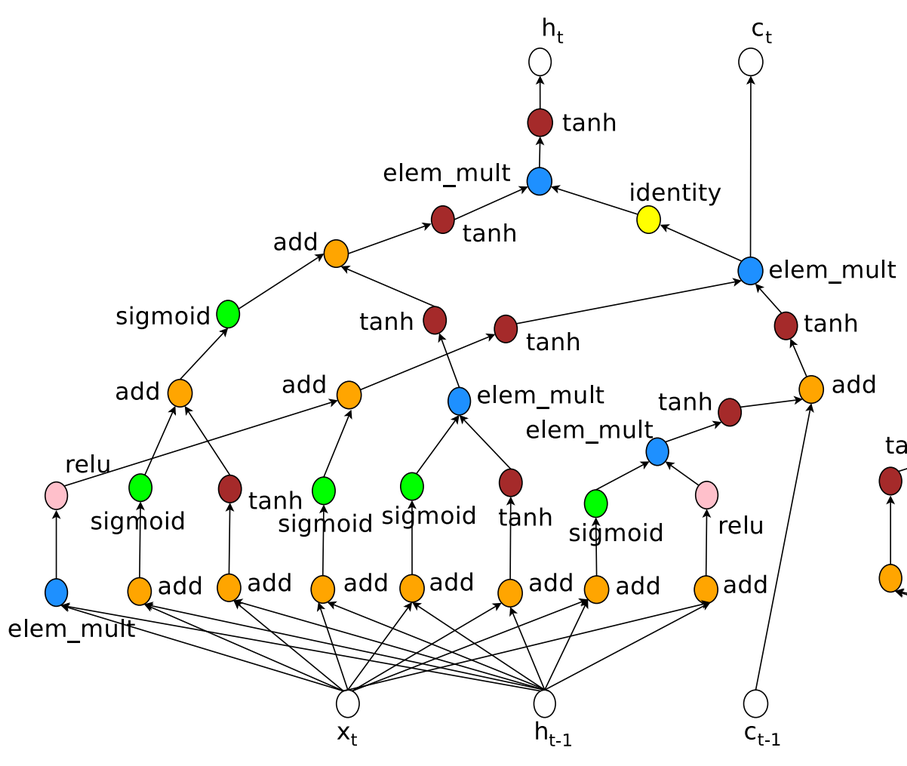




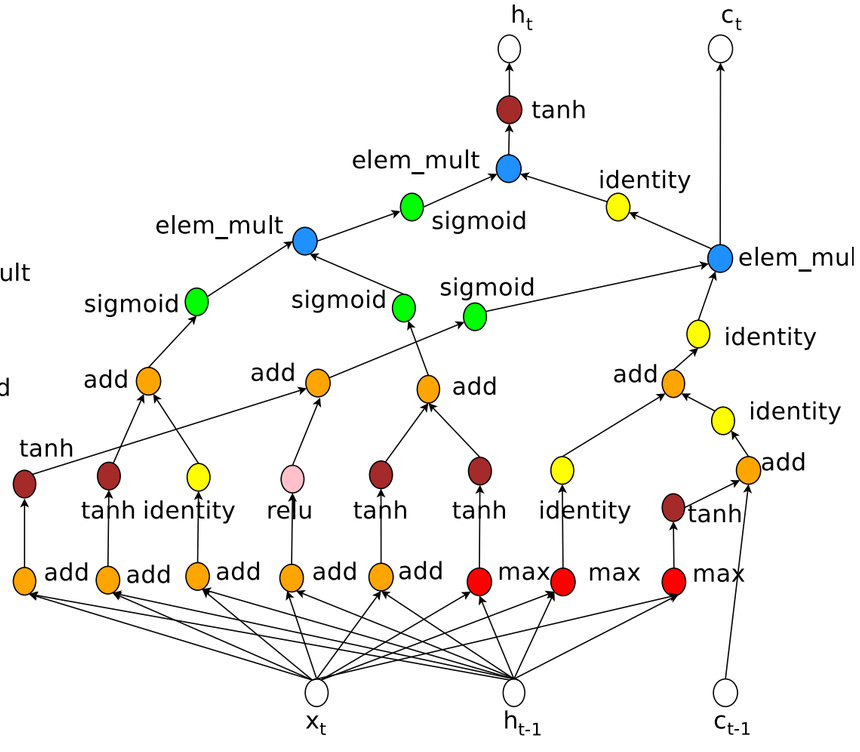
LSTM cell



Found cell (no max or sin)



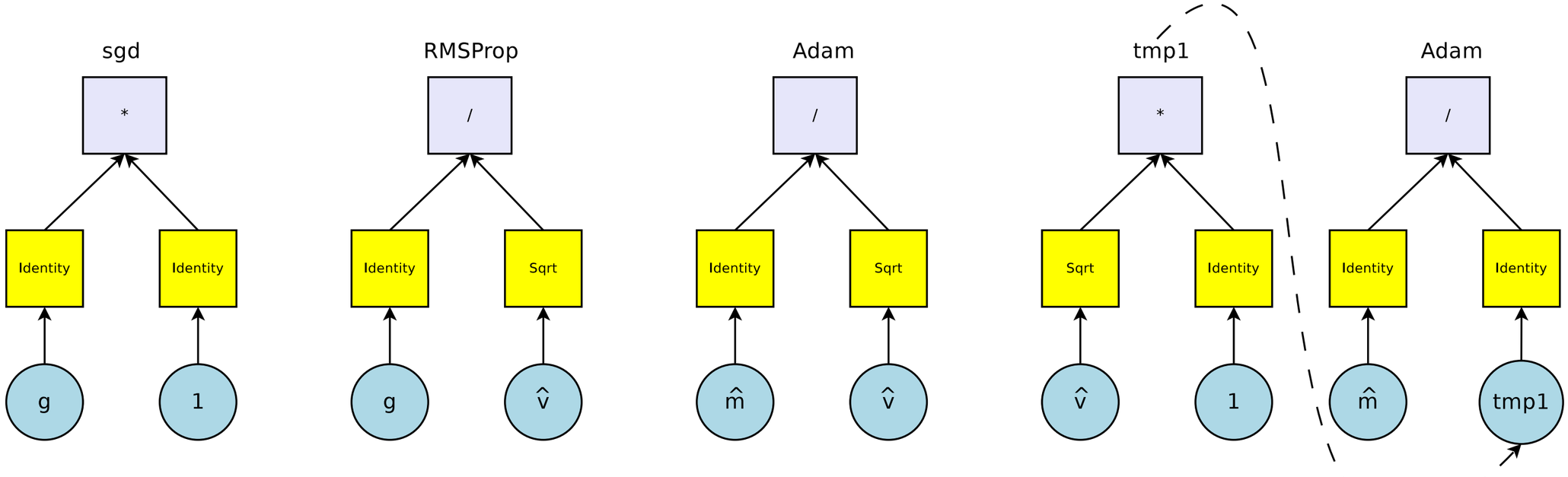
Found cell (max, sin included)



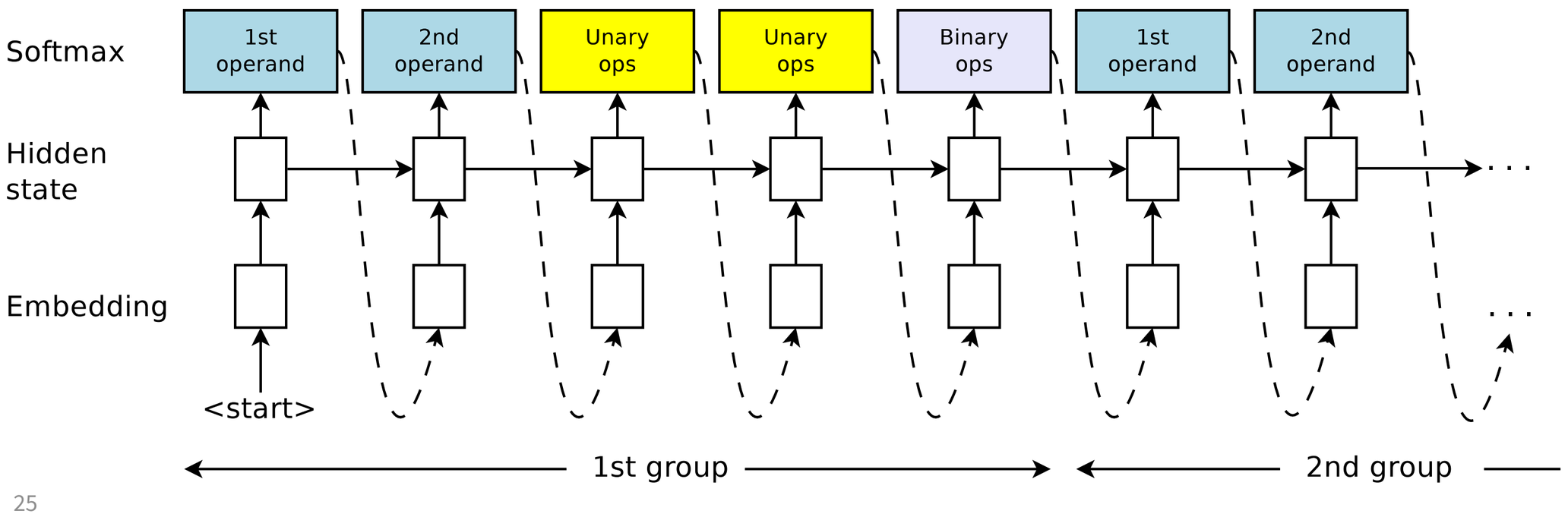
via Zoph and Le (2017)

Meta-learning: Optimization

Domain-specific language (DSL) for optimizers

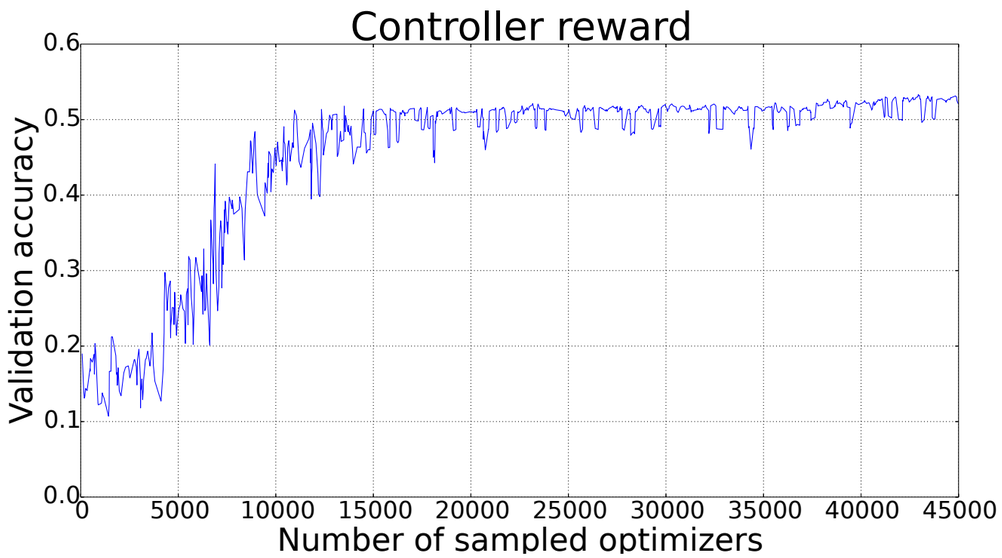


RNN controller outputs optimizer as string



via Zoph and Le (2017)

Results: Learning Optimizers

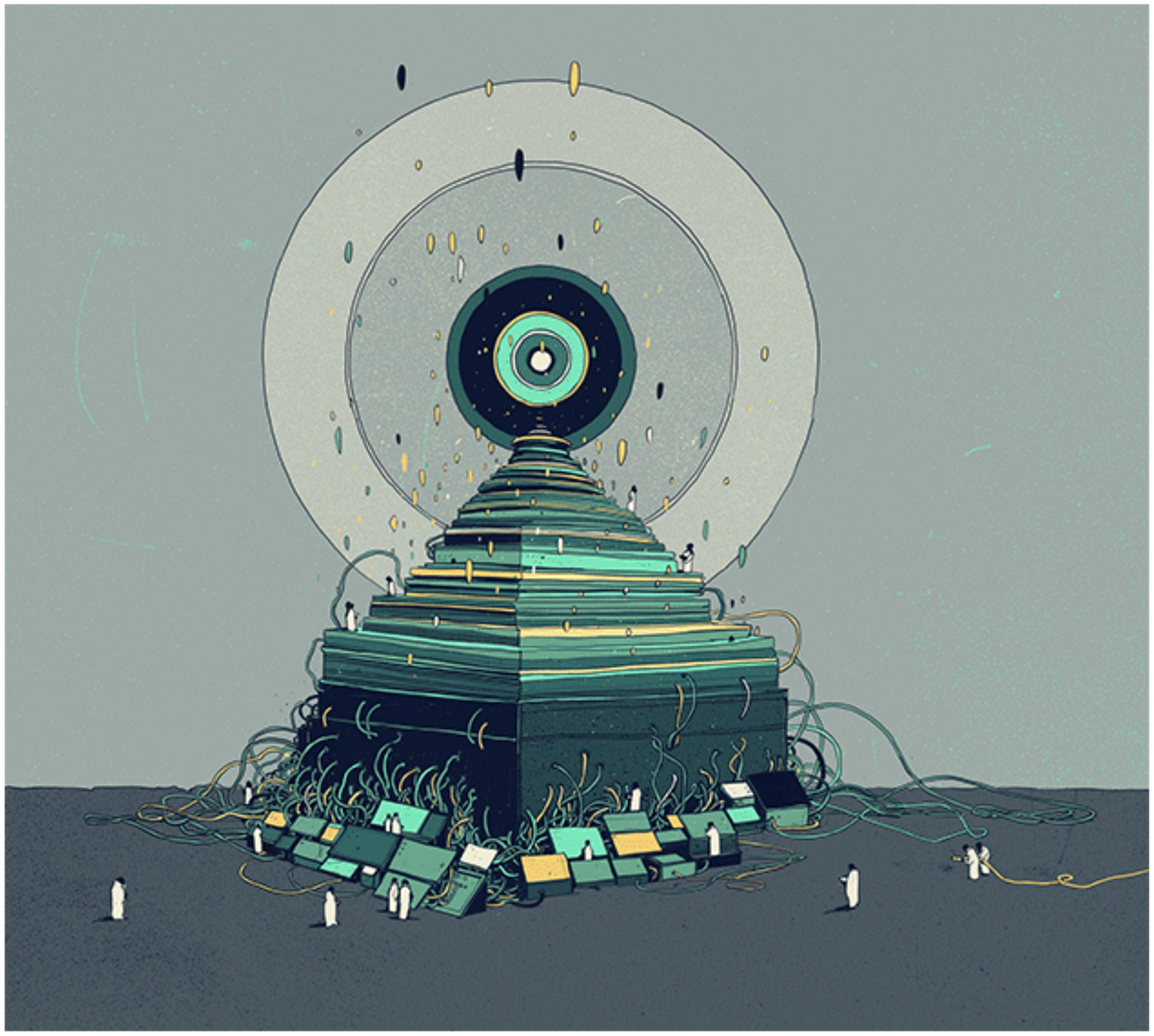


Optimizer	Final Val	Final Test	Best Val	Best Test
SGD	92.0	91.8	92.9	91.9
Momentum	92.7	92.1	93.1	92.3
Adam	90.4	90.1	91.8	90.7
RMSProp	90.7	90.3	91.4	90.3
$\hat{m} * (1 + \epsilon) * sigmoid(10^{-4} w)$	90.6	90.6	93.1	92.2
$\text{sign}(m) * \sqrt{\lvert g \rvert}$	92.2	91.8	92.9	92.2
$\text{sign}(g) * \text{sign}(m) * \hat{m}$	91.2	91.0	92.4	91.3
$\text{sign}(m) * \sqrt{\lvert g \rvert}$	91.7	91.1	92.3	91.6
$(1 + \text{sign}(g) * \text{sign}(m)) * \text{sign}(g)$	91.3	90.4	91.9	91.1
$(1 + \text{sign}(g) * \text{sign}(m)) * \text{sign}(m)$	91.0	90.6	92.0	90.8
$\text{sign}(g) * \sqrt{\lvert \hat{m} \rvert}$	90.7	90.6	91.5	90.6
$\sqrt{\lvert g \rvert} * \hat{m}$	92.0	90.9	93.6	93.1
$\sqrt{\lvert g \rvert} * g$	92.6	91.9	93.2	92.3
$(1 + \text{sign}(g) * \text{sign}(m)) * \hat{m}$	91.8	91.3	92.6	91.8
$(1 + \text{sign}(g) * \text{sign}(m)) * \text{RMSProp}$	92.0	92.1	92.9	92.4
$(1 + \text{sign}(g) * \text{sign}(m)) * \text{Adam}$	91.2	91.2	92.2	91.9
$[e^{\text{sign}(g) * \text{sign}(m)} + \text{clip}(g, 10^{-4})] * g$	92.5	92.4	93.8	93.1
$\text{clip}(\hat{m}, 10^{-4}) * e^{\hat{v}}$	93.5	92.5	93.8	92.7
$\hat{m} * e^{\hat{v}}$	93.1	92.4	93.8	92.6
$g * e^{\text{sign}(g) * \text{sign}(m)}$	93.1	92.8	93.8	92.8
$\text{drop}(g, 0.3) * e^{\text{sign}(g) * \text{sign}(m)}$	92.7	92.2	93.6	92.7
$\hat{m} * e^{g^2}$	93.1	92.5	93.6	92.4
$\text{drop}(\hat{m}, 0.1) / (e^{g^2} + \epsilon)$	92.6	92.4	93.5	93.0
$\text{drop}(g, 0.1) * e^{\text{sign}(g) * \text{sign}(m)}$	92.8	92.4	93.5	92.2
$\text{clip}(\text{RMSProp}, 10^{-5}) + \text{drop}(\hat{m}, 0.3)$	90.8	90.8	91.4	90.9
$\text{Adam} * e^{\text{sign}(g) * \text{sign}(m)}$	92.6	92.0	93.4	92.0
$\text{Adam} * e^{\hat{m}}$	92.9	92.8	93.3	92.7
$g + \text{drop}(\hat{m}, 0.3)$	93.4	92.9	93.7	92.9
$\text{drop}(\hat{m}, 0.1) * e^{g^3}$	92.8	92.7	93.7	92.8
$g - \text{clip}(g^2, 10^{-4})$	93.4	92.8	93.7	92.8
$e^g - e^{\hat{m}}$	93.2	92.5	93.5	93.1
$\text{drop}(\hat{m}, 0.3) * e^{10^{-3} w}$	93.2	93.0	93.5	93.2

via Zoph and Le (2017)

Explainability in AI

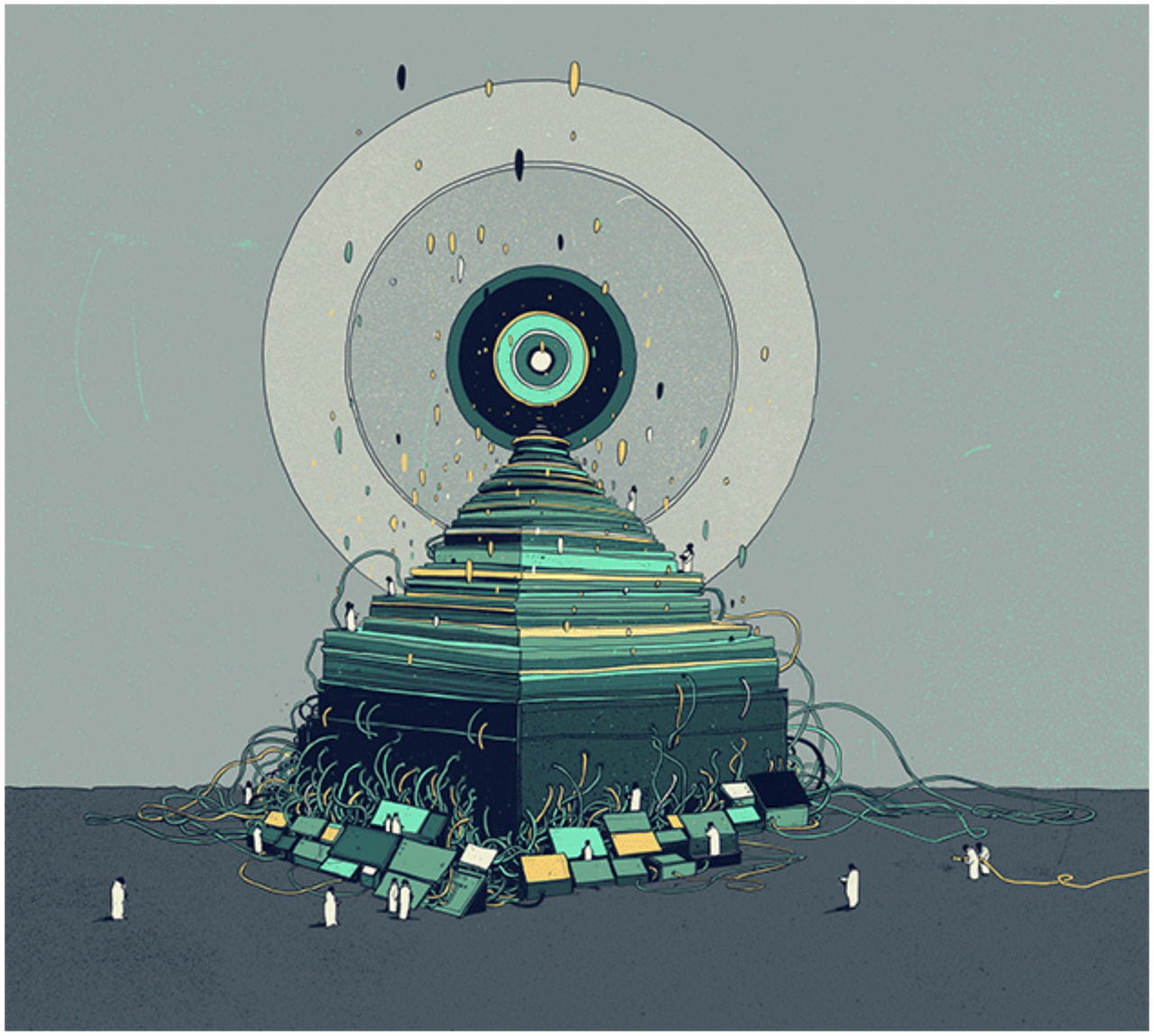
- The most accurate models in Machine Learning today are the most opaque, non-intuitive, and difficult for people to understand
- Explainable AI systems are necessary if users are to understand and appropriately trust their outputs



via nature.com

Explainability in AI

- The most accurate models in Machine Learning today are the most opaque, non-intuitive, and difficult for people to understand
- Explainable AI systems are necessary if users are to understand and appropriately trust their outputs



via nature.com

EU “right to explanation”

CADE METZ BUSINESS 07.11.16 07:00 AM

ARTIFICIAL INTELLIGENCE IS SETTING UP THE INTERNET FOR A HUGE CLASH WITH EUROPE

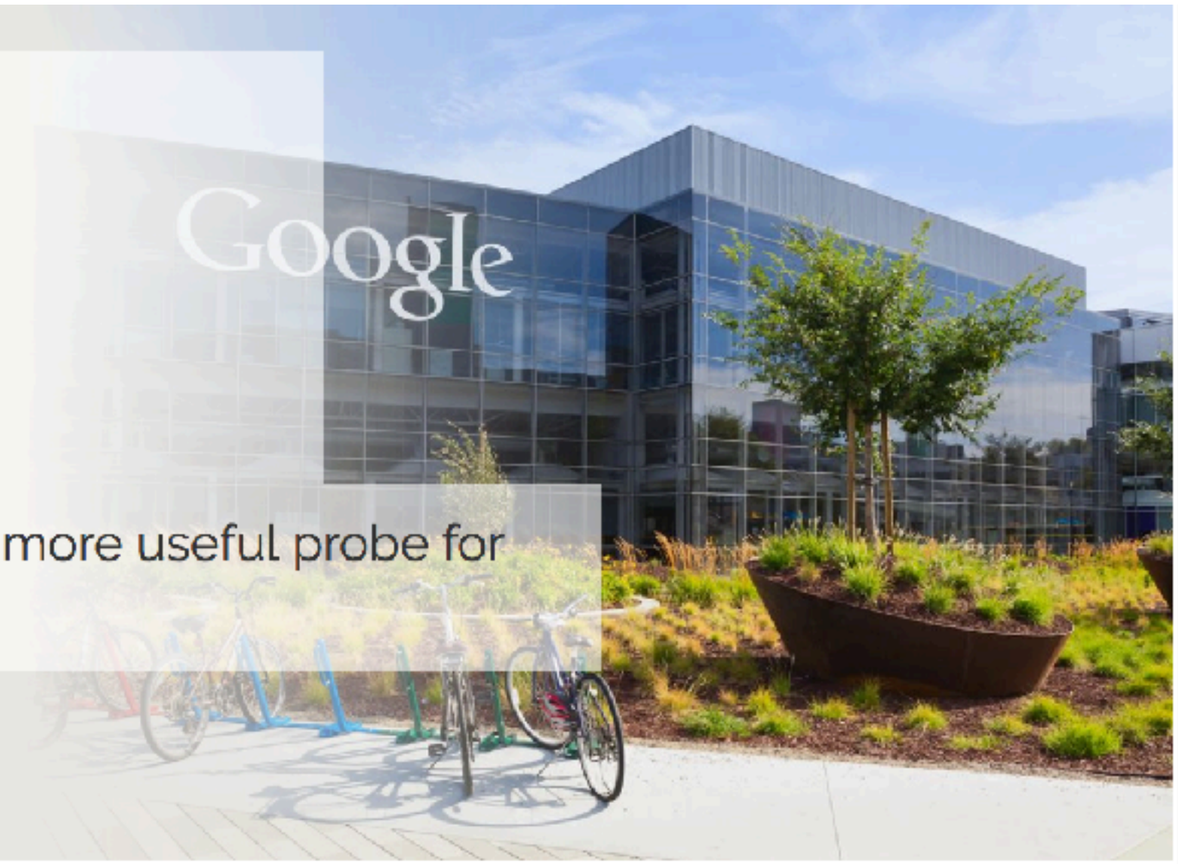
GETTY IMAGES

via wired.com

Value?

Google's research chief questions value of 'Explainable AI'

Peter Norvig says output of machine learning systems a more useful probe for fairness



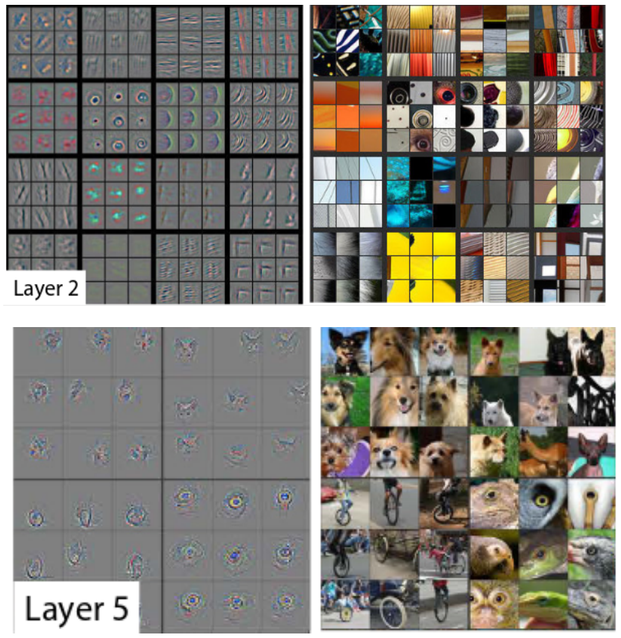
George Nott (Computerworld)
23 June, 2017 11:53

16 Oct 2017 / 29
RBC · New Paradigms / G Taylor

via Computerworld

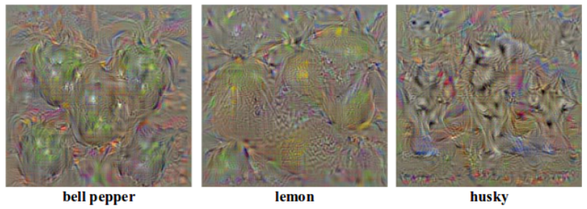
Previous Work on Visualization

Deconvolution

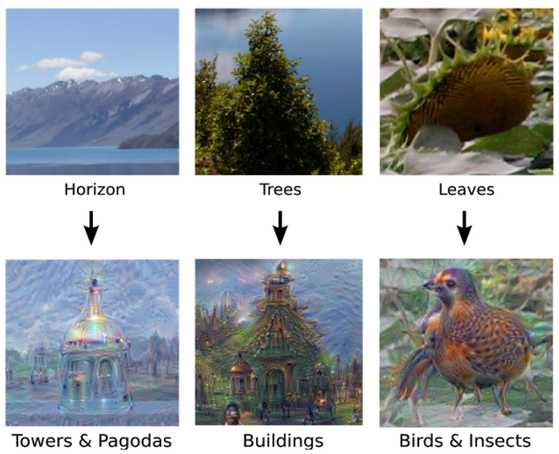


(Zeiler and Fergus 2014)

Backpropagation

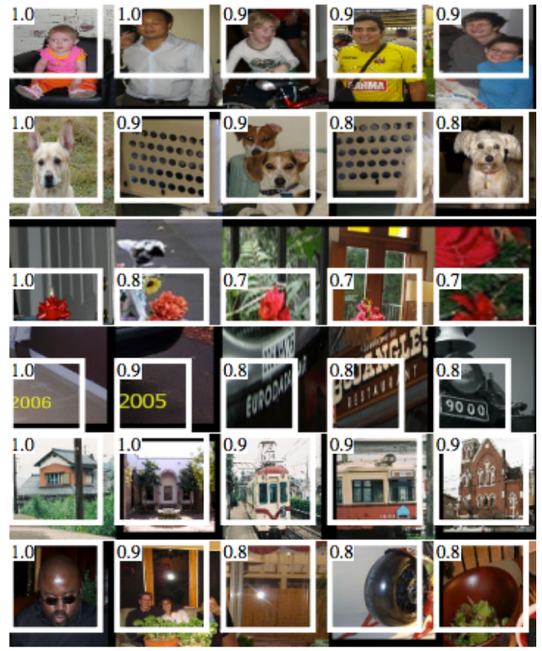


(Simonyan et al. 2015)



Inceptionism (Google Blog)

Top activated images



(Girshick et al. 2014)

via Bau et al. 2017

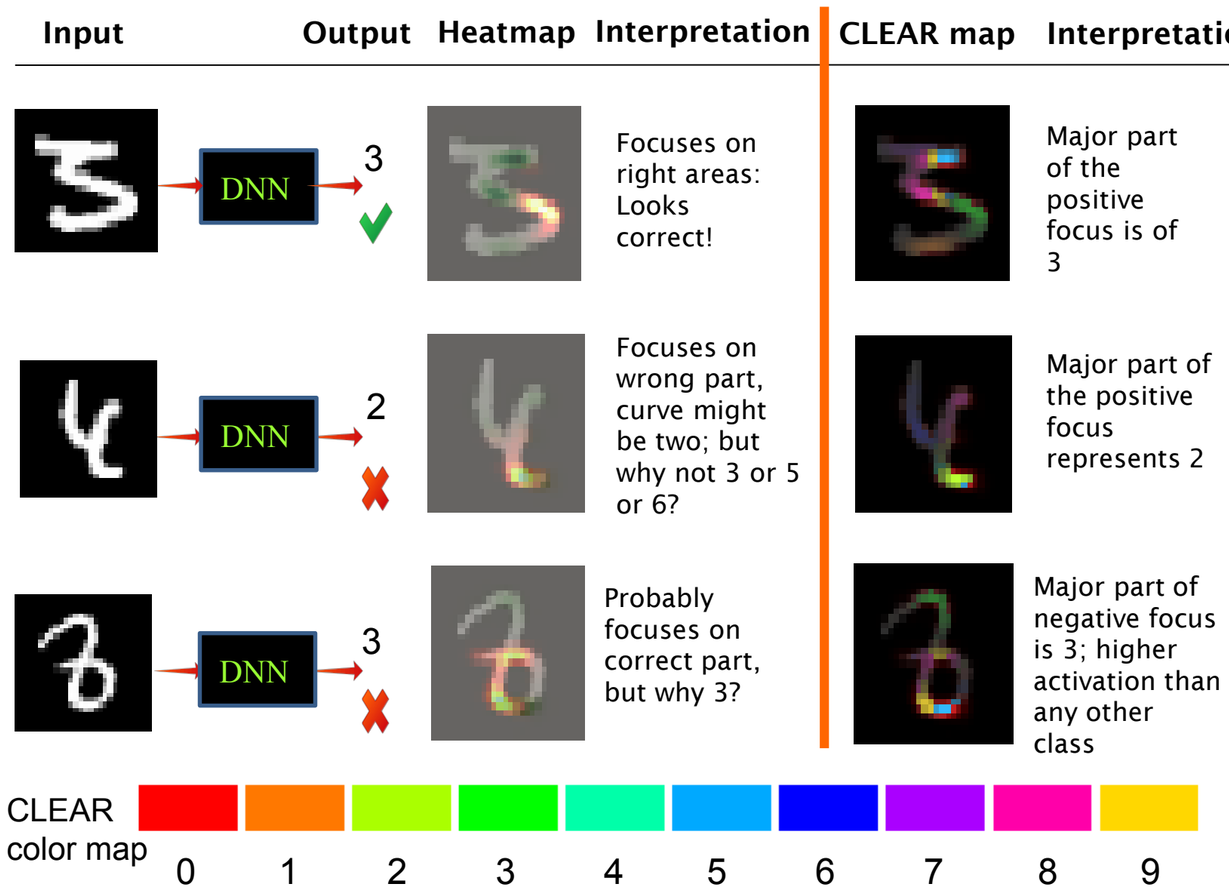
CLass-Enhanced Attentive Response (CLEAR)

Devinder Kumar

Alex Wong

CLEAR provides:

- attentive regions of interest
- corresponding attentive levels
- dominant class for each attentive region



Kumar, Wong and Taylor (2017)
"A Class-Enhanced Attentive Response Approach to Understanding Deep Neural Networks "

Network Dissection

Network Dissection (Bau et al. 2017) is a method that investigates 3 questions:

1. What is a disentangled representation, and can its factors be quantified and detected?

2. Do interpretable hidden units reflect a special alignment of feature space, or are interpretations a myth?

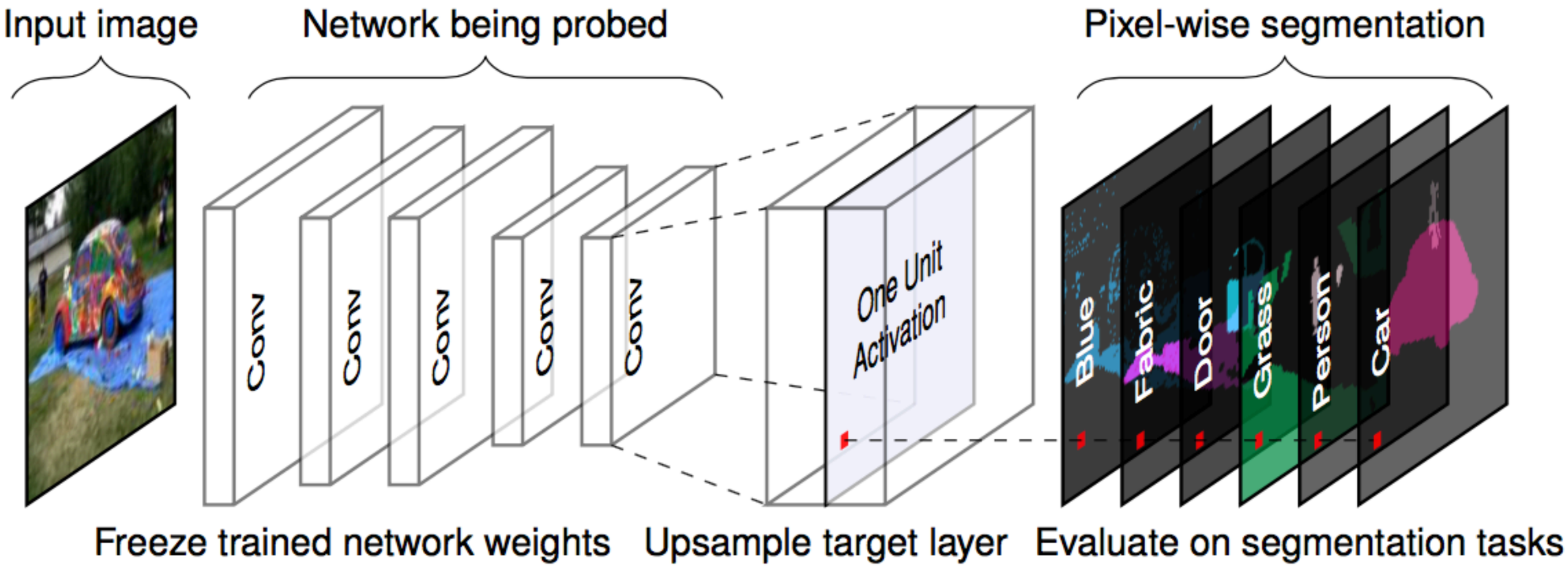
3. What conditions in training lead to greater or less entanglement?

via Bau et al. 2017

Quantifying Interpretability

Network Dissection is a method for quantifying the interpretability of individual units in a deep CNN (it answers question #1)

It works by measuring the alignment between unit response and a set of concepts drawn from a broad and dense segmentation set



via Bau et al. 2017

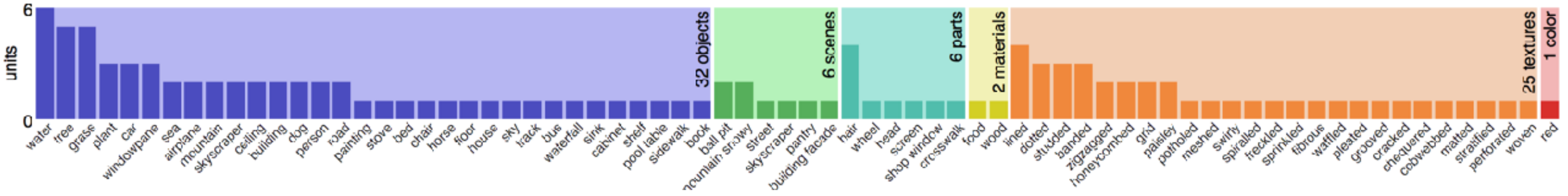
From Units to Concepts

Unit 1

Top activated images

Lamp

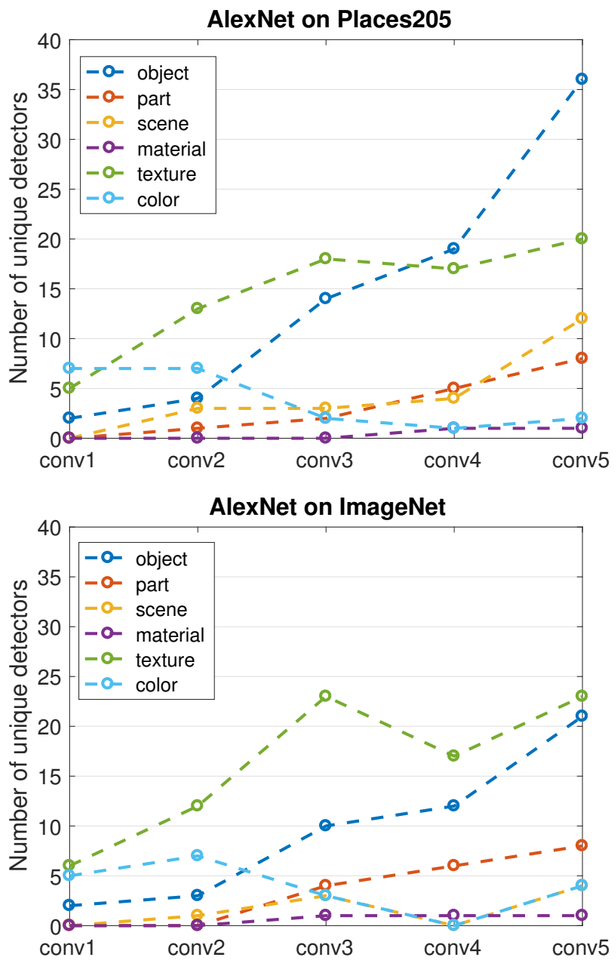
IOU = 0.12

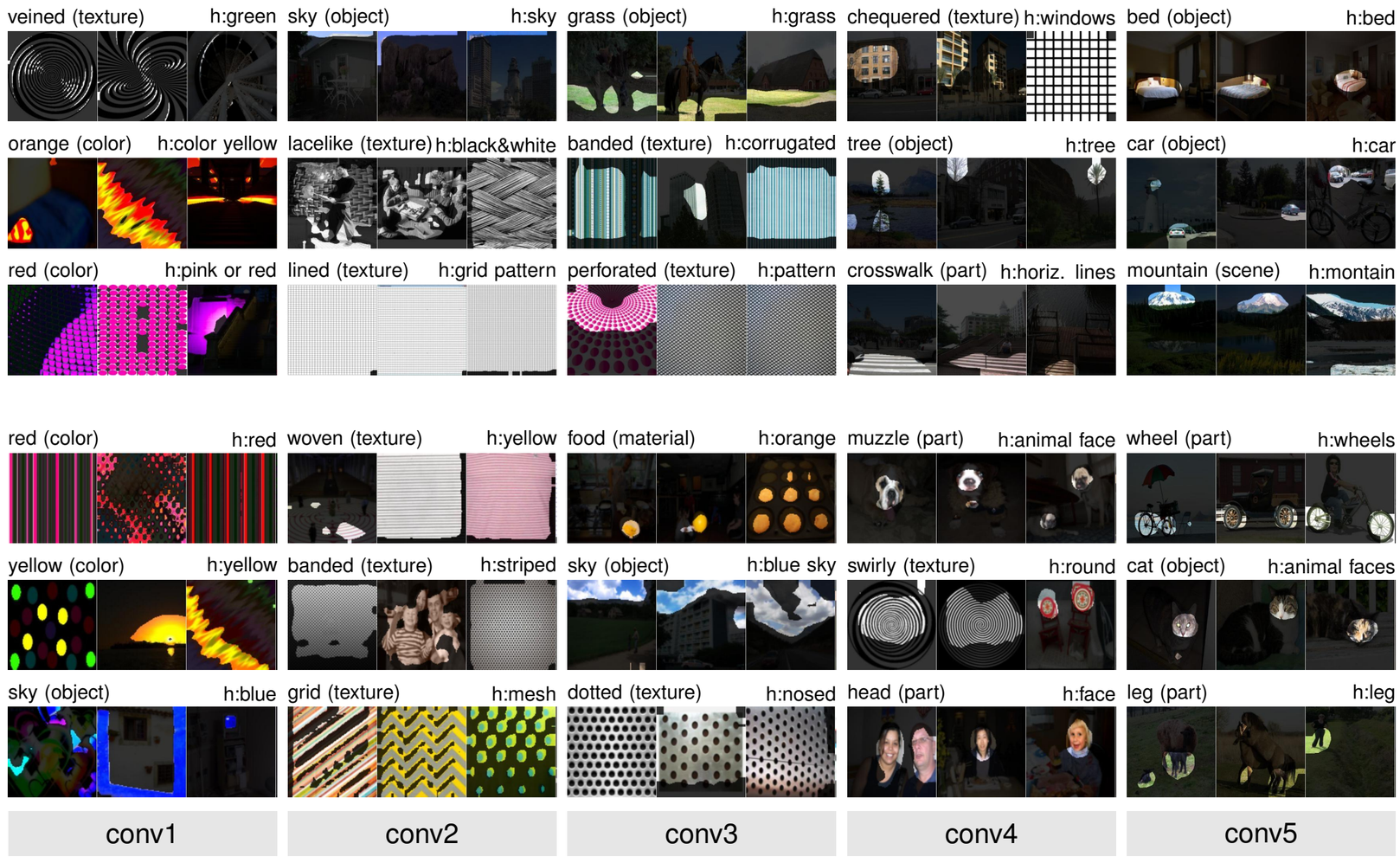


256 units of an AlexNet conv5 trained on the Places dataset

via Bau et al. 2017

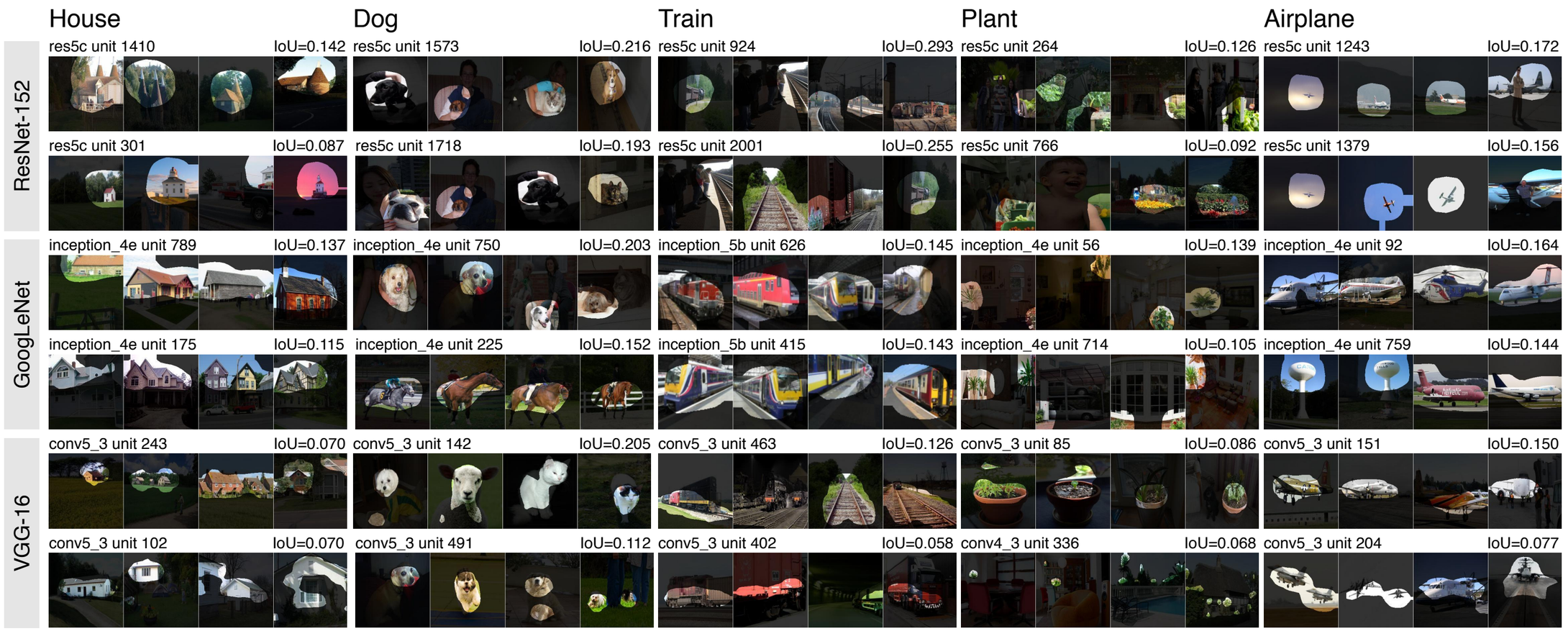
Interpretability Across Layers





via Bau et al. 2017

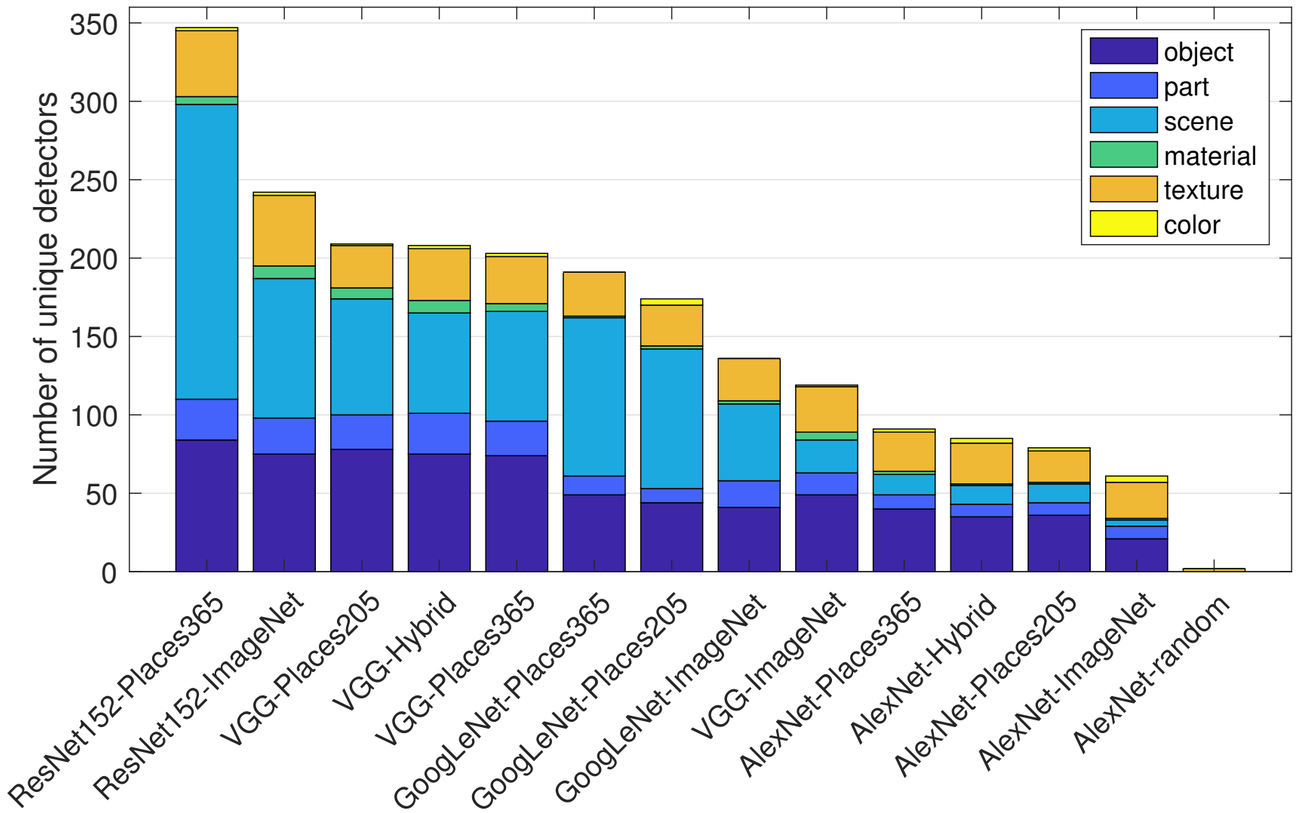
Interpretability Across Models



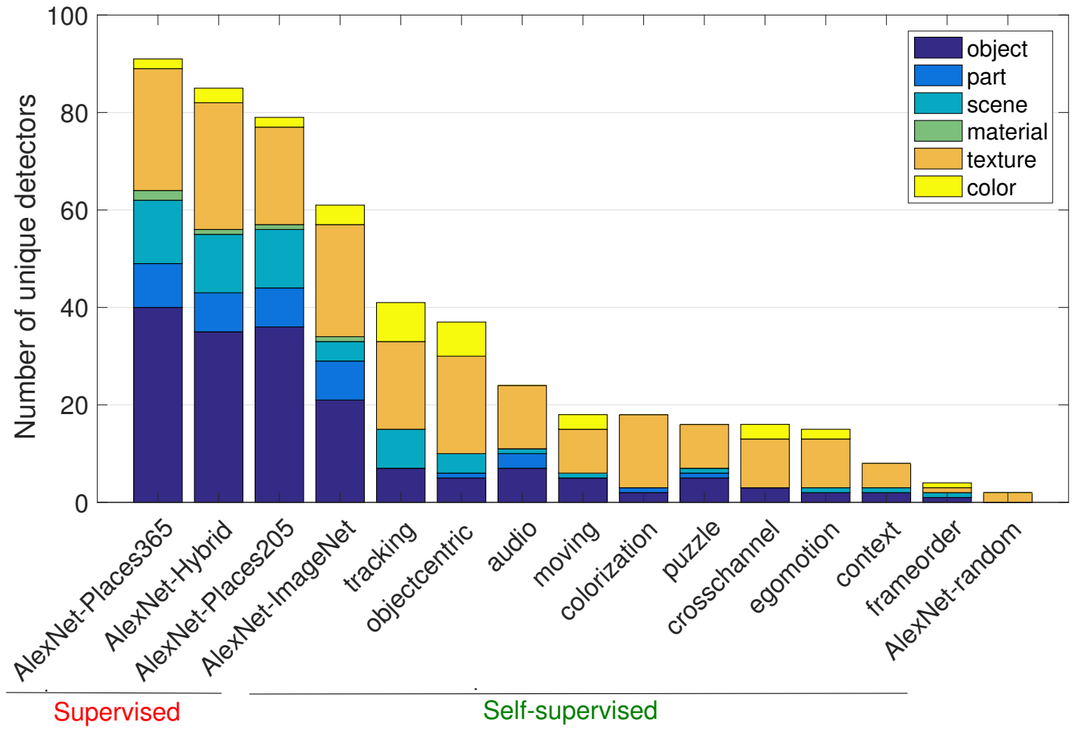
via Bau et al. 2017

Interpretability Across Models (2)

Across architectures



Across learning strategies



via Bau et al. 2017