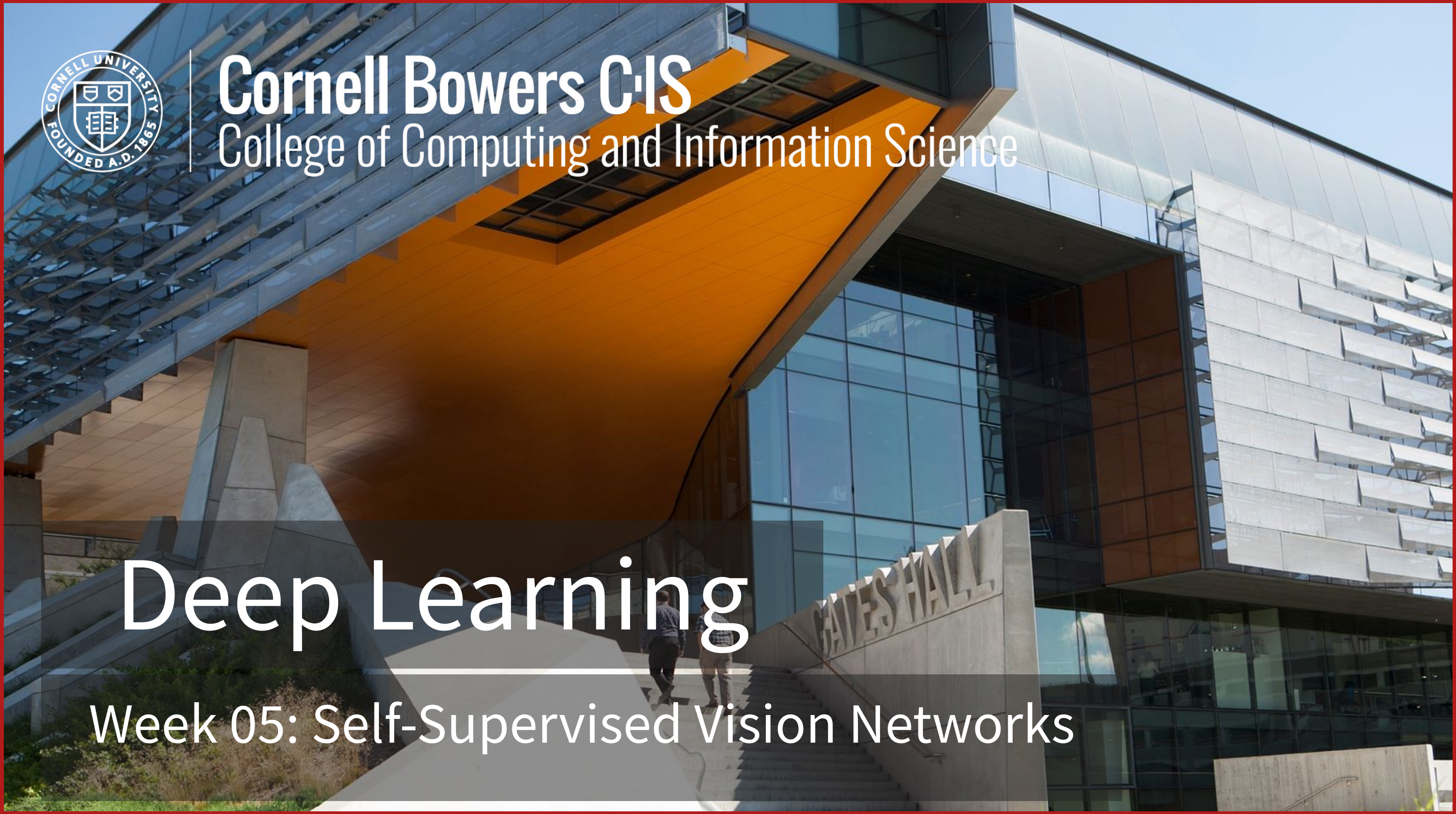
Cornell Bowers C·IS
College of Computing and Information Science

Deep Learning

Week 05: Self-Supervised Vision Networks



Cornell Bowers C·IS

Logistics

- HW3 is out
- We have a feedback form (due Friday, March 8th)
- Project proposal due Thursday, March 7th

Project

- Aim: to get hands on experience with implementing modern deep learning methods
- To be completed in groups of 2-3
- Find a recent deep learning research paper
- Reproduce a specific result from the paper
  - Need to implement yourself!
- It’s ok if open-source implementations exist
  - But you can’t use them!

Cornell Bowers C·IS

Project Proposal

A page long project proposal due **March 7**. It should contain the following:

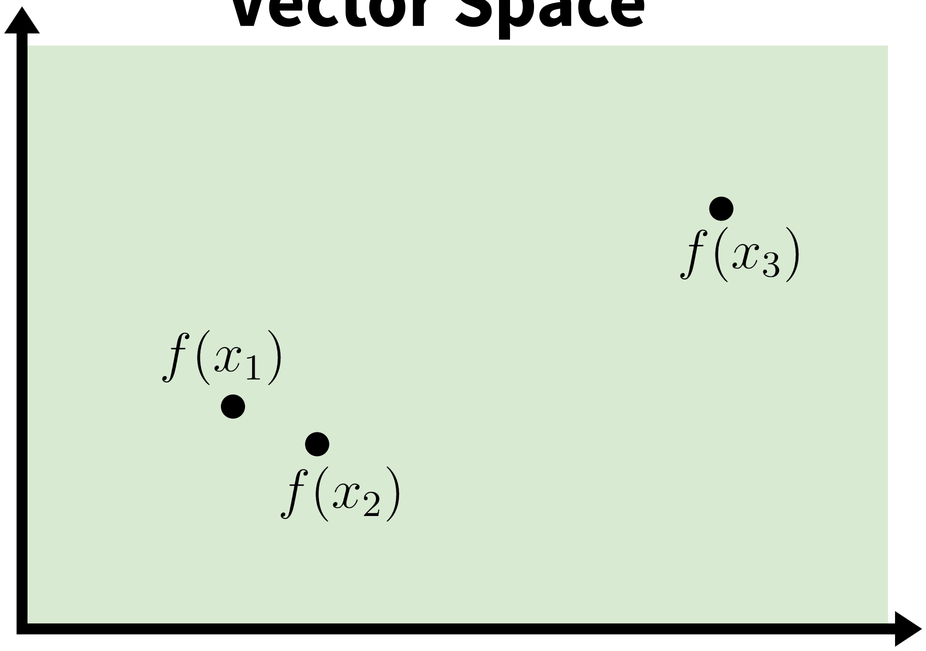
- Paper selection:
  - Title, authors, and publication venue of the chosen paper
  - Brief summary of the chosen paper
  - Brief justification of why you choose this paper
- Result Selection
  - Tell us which result you want to replicate
- Re-implementation Plan
  - Describe architecture, method, and metrics
  - Details about how much compute and time is required to replicate results
- **Detailed instructions on canvas**

Cornell Bowers CIS

$f(x)$ = bag of words

x_1	x_2	x_3
The	The	I
tastiest	best	enjoy
fruits	animals	petting
are	are	baby
oranges	puppies	dogs

Vector Space



Cornell Bowers CIS

$f(x)$ = word embedding

"orange" x_1

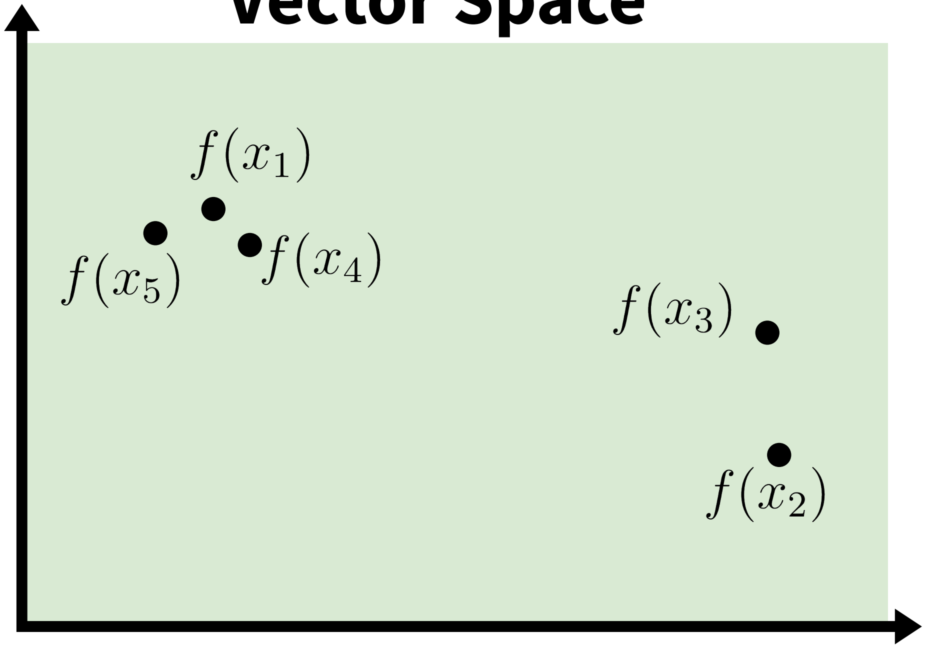
"purple" x_4

"puppies" x_2

"apple" x_5

"dogs" x_3

Vector Space



Cornell Bowers C·IS

Semantically different:
puppy vs. cow

Structurally similar:
black and white animal, grass

x_1

x_2

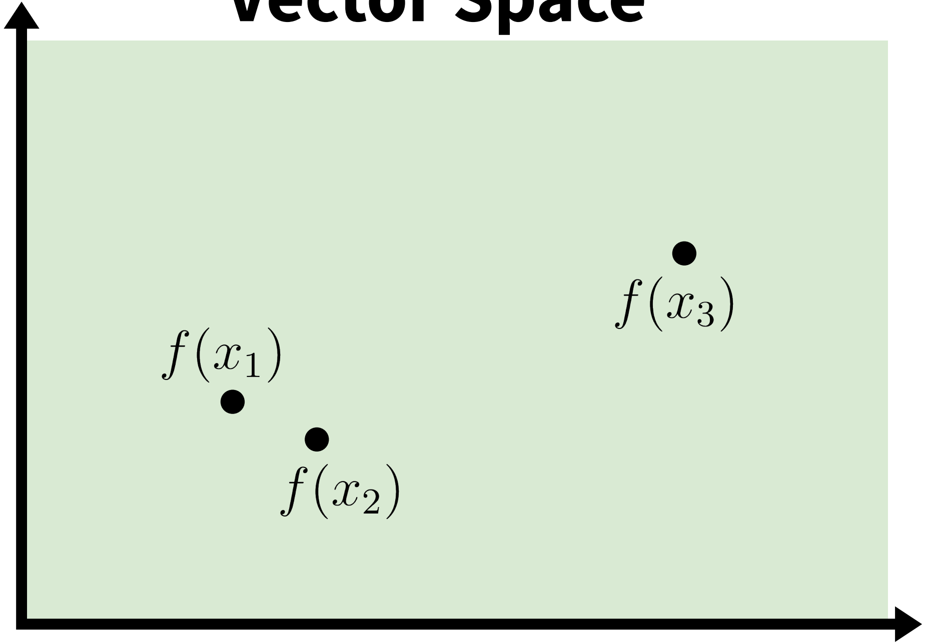
Structurally different:
hands, different backgrounds

Semantically similar:
Bernese puppies

x_3

$f(x)$ = raw pixels

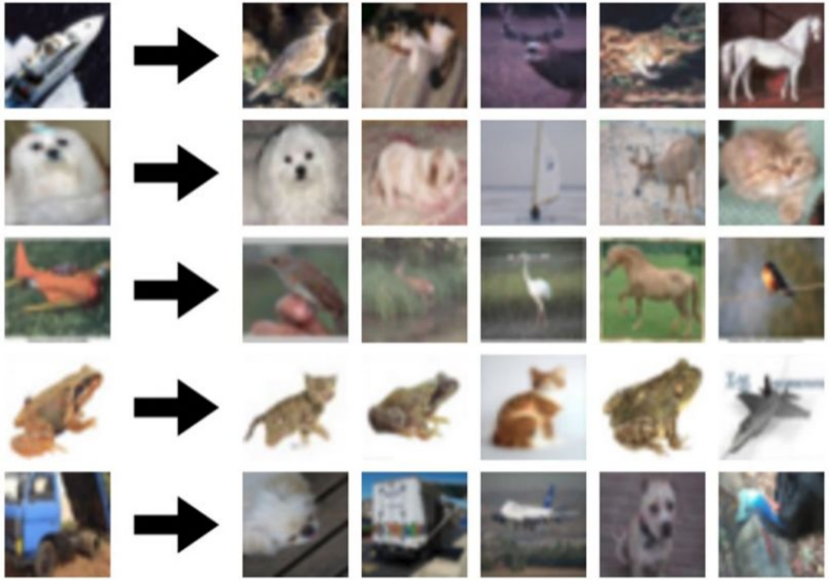
Vector Space



Cornell Bowers C·IS

Pixel-Space: Nearest Neighbors

- Dominated by shallow similarities
  - Background, etc.
- Poor semantic alignment



Cifar-10 Example
(http://cs231n.stanford.edu/slides/2023/lecture_13.pdf)

Cornell Bowers C·IS

Semantically different:
puppy vs. cow

Structurally similar:
black and white animal, grass

x_1

x_2

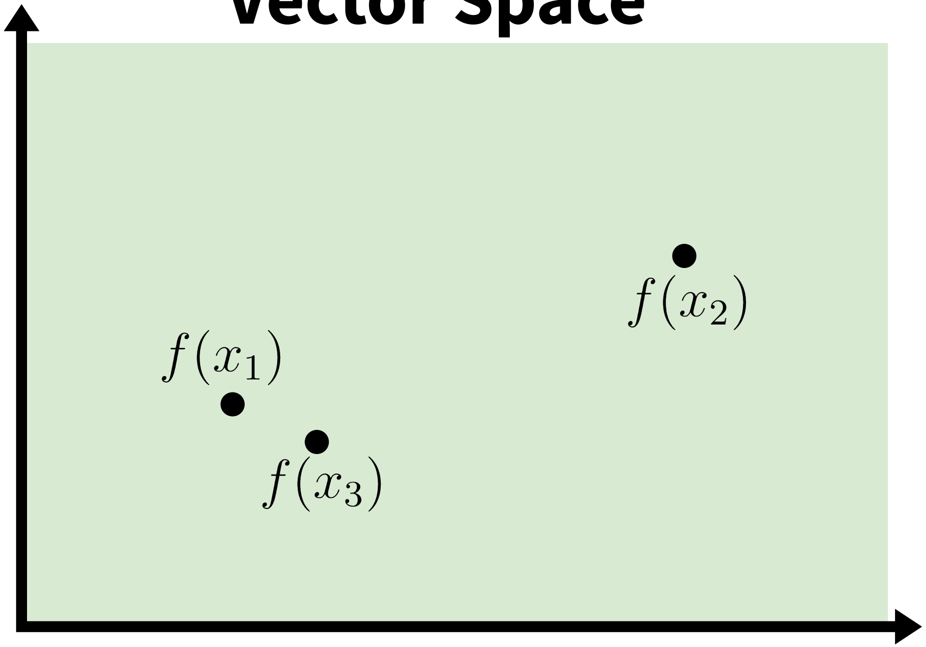
Structurally different:
hands, different backgrounds

Semantically similar:
Bernese puppies

x_3

$f(x)$ = classification network

Vector Space

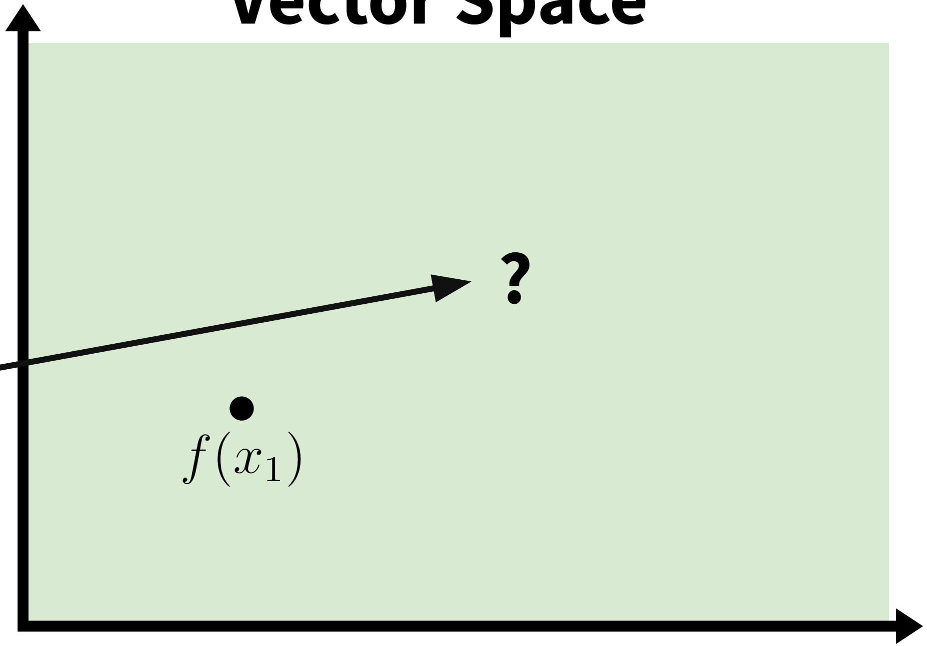


Cornell Bowers C·IS

$f(x)$ = classification network

How does the network know that these should be mapped to similar space?

Vector Space



x_1

x_2

Cornell Bowers C·IS

$f(x)$ = classification network

How does the network know that these should be mapped to similar space?

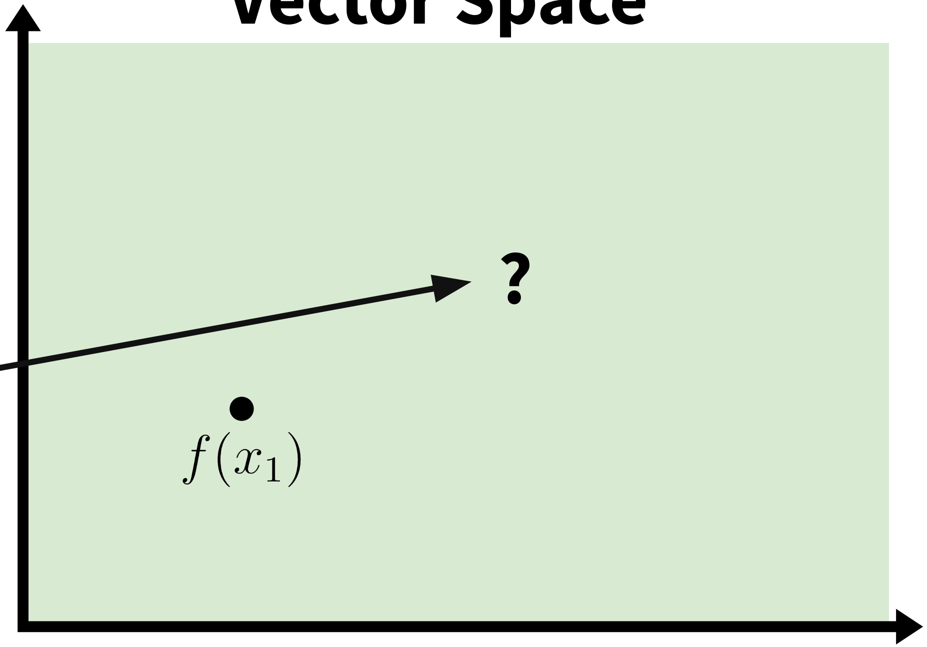
Class "puppy"

x_1

Class "puppy"

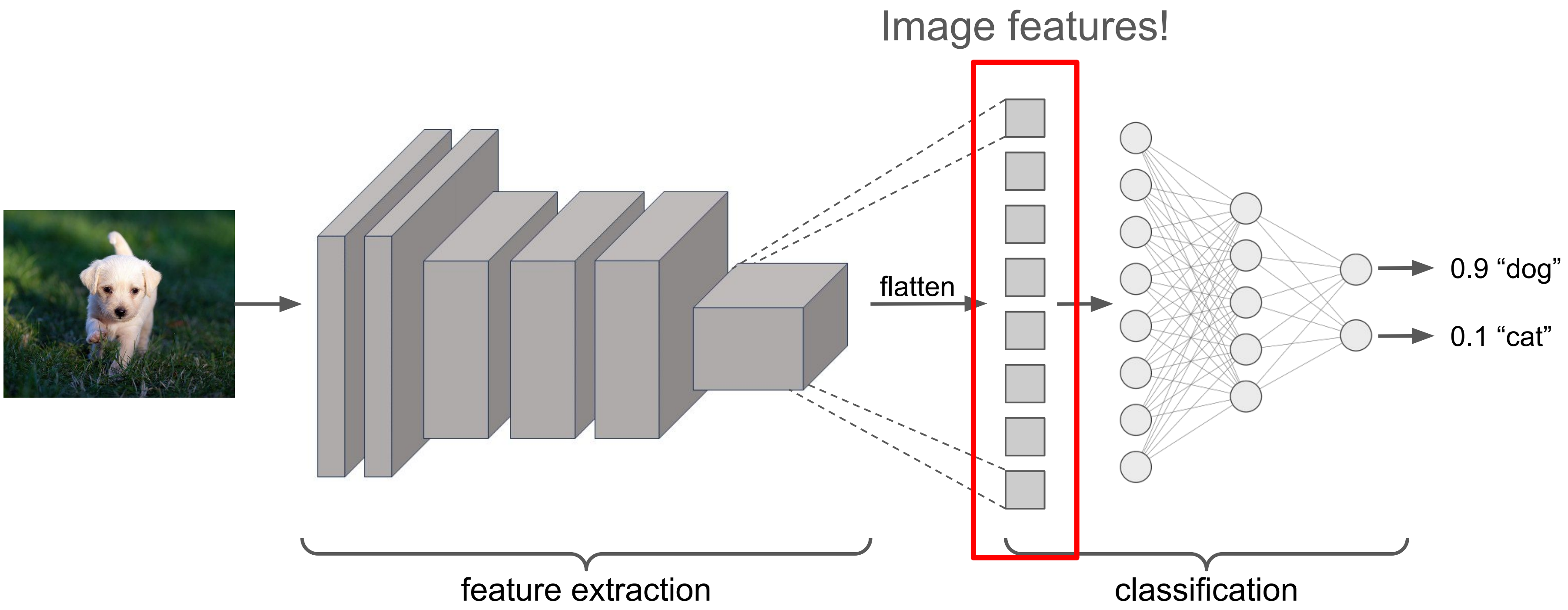
x_2

Vector Space



Cornell Bowers C·IS

Image Classification



Cornell Bowers C·IS

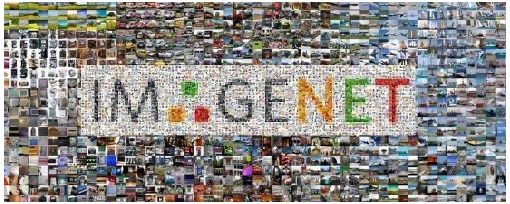
Neural Net Features: Nearest Neighbors

- Image classification features work really well!
- Strong semantic alignment
- More robust to shallow variations

ImageNet Classification with Deep Convolutional Neural Networks by Krizhevsky et al.

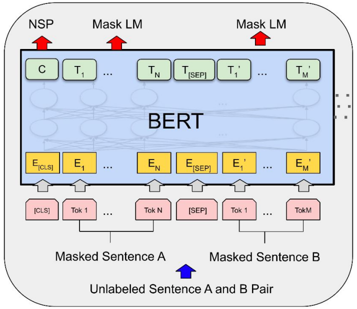
Cornell Bowers C·IS

Pretraining: Train a general purpose model on lots of data, and later customize it for more specific tasks



CV: **Imagenet**

NLP: **BERT**



Pre-training

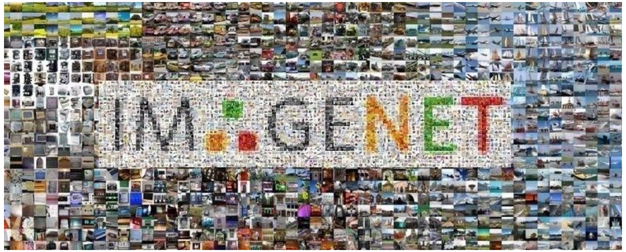
Vector Space

Already have a very well-defined vector space

Cornell Bowers C·IS

Image Pretraining

First, train on a large, diverse dataset so that our model learns to extract robust image features



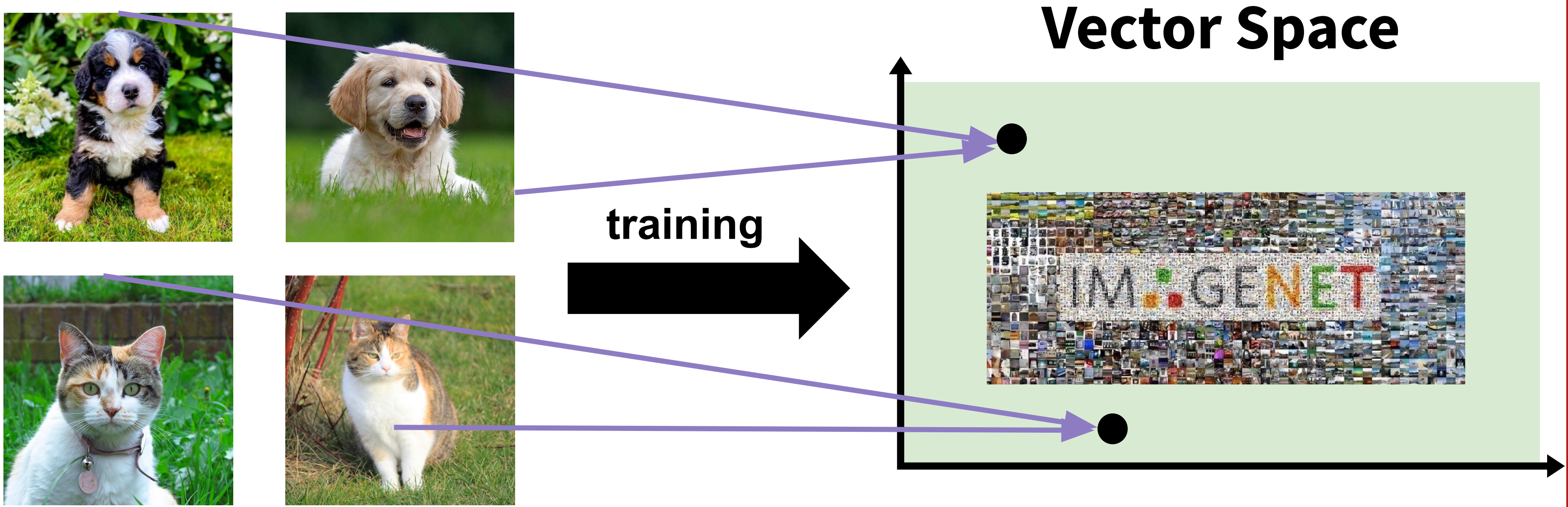
training →

Vector Space

Cornell Bowers C·IS

Fine-tuning

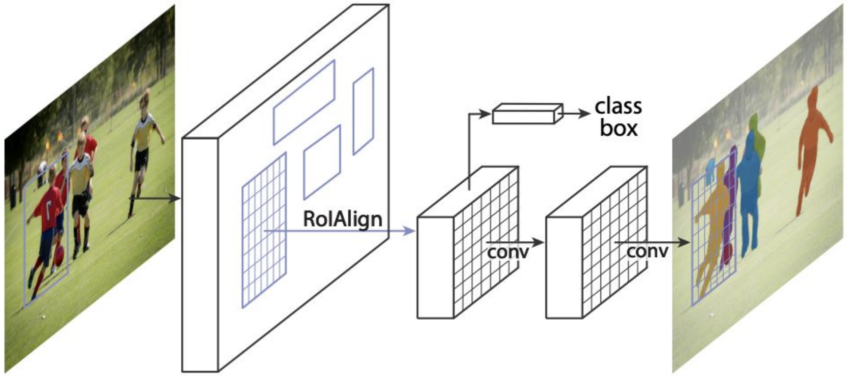
Then, finetune for a specific task

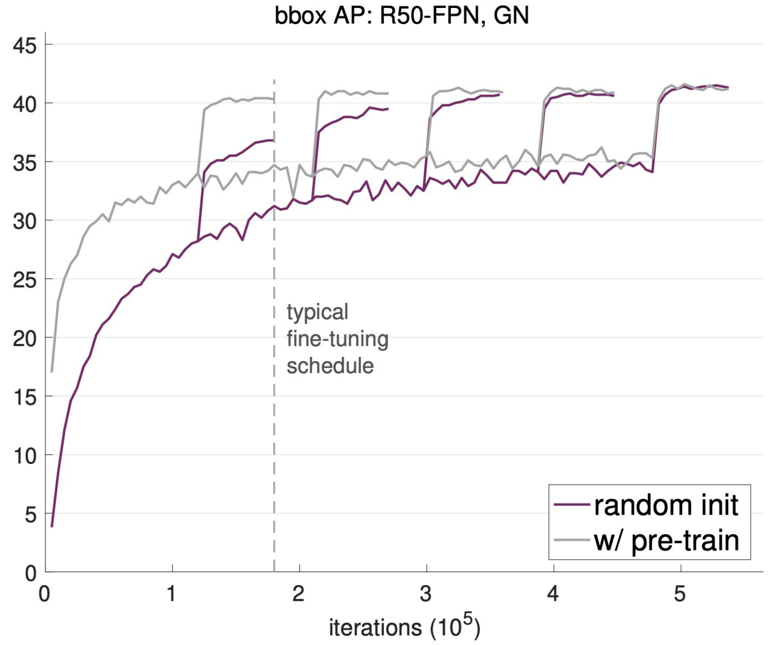


Cornell Bowers C·IS

Pre-train then Fine-tune

- Use image classification backbone as a feature extractor for other vision tasks
    - E.g. Instance segmentation
- Significantly accelerates training
    - Random init requires much longer training





Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." Advances in neural information processing systems 28 (2015).
He, Kaiming, Ross Girshick, and Piotr Dollár. "Rethinking imagenet pre-training." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019.

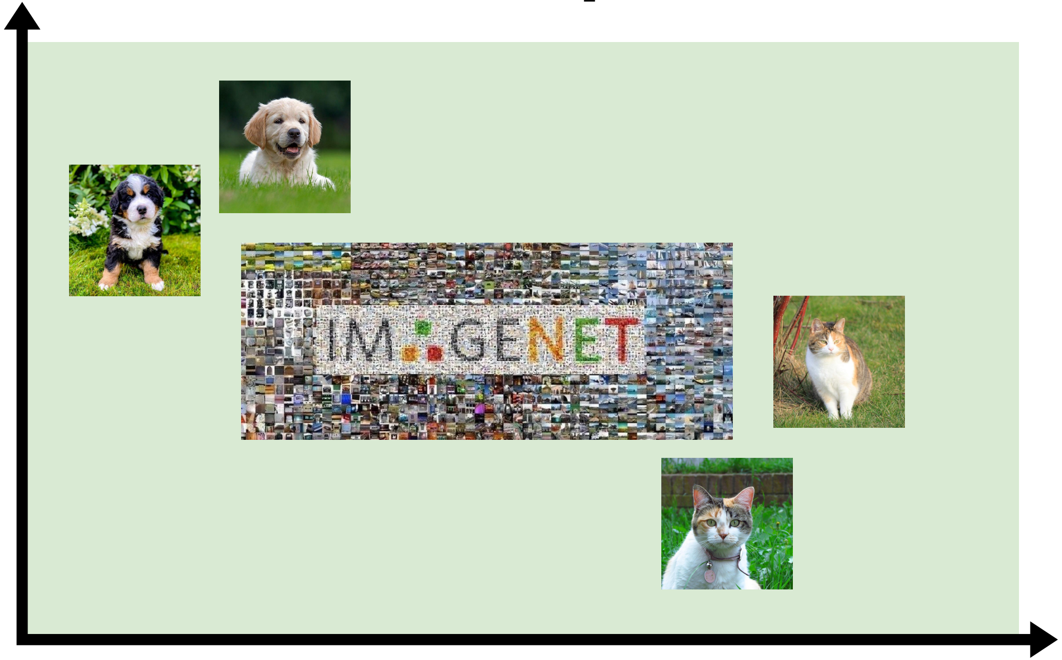
Cornell Bowers C·IS

Few Shot Learning

Adapt to variations within known classes, with LIMITED labeled training data

- **We've only seen a few puppies and a few kittens, but a lot of other pretrained data**

Vector Space



Cornell Bowers C·IS

Potential Problems?

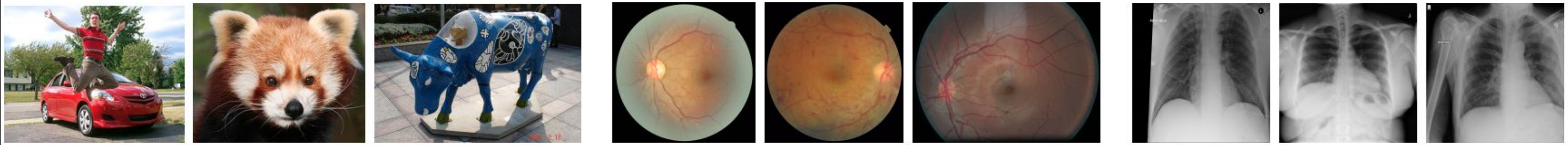


Figure 1: Example images from the IMAGENET, the *retinal fundus photographs*, and the CHEXPERT datasets, respectively. The fundus photographs and chest x-rays have much higher resolution than the IMAGENET images, and are classified by looking for small local variations in tissue.

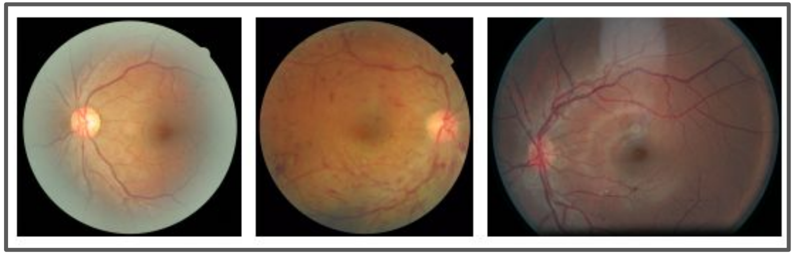
Raghu, Maithra, et al. "Transfusion: Understanding transfer learning for medical imaging." Advances in neural information processing systems 32 (2019).

Cornell Bowers C·IS

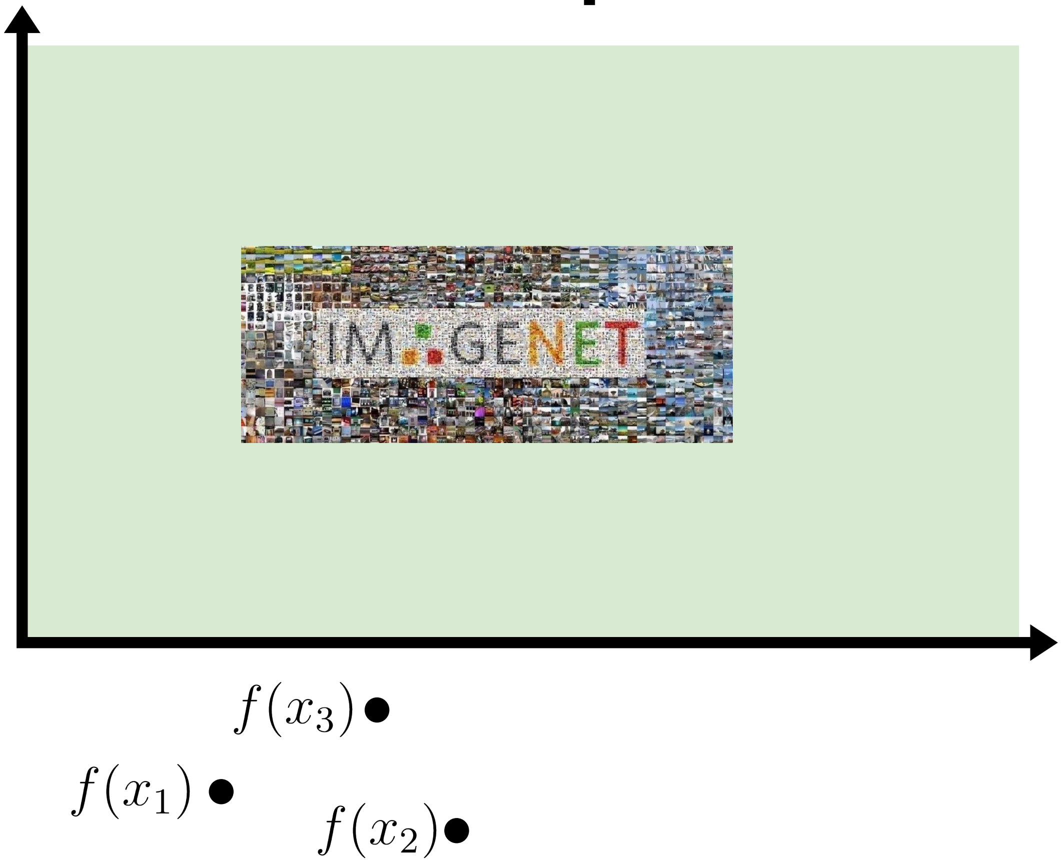
Transfer Learning

Images may be out-of-distribution from the training data

x_1 x_2 x_3

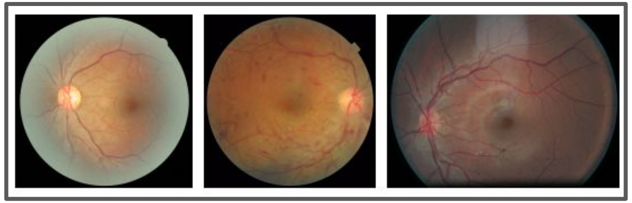


Vector Space



Potential Problems?

- Classify diabetic retinopathy in retinal photographs
- Introduce classification simple architecture
  - Sequence of: Convolution, Batchnorm, ReLU (CBR)



Dataset	Model Architecture	Random Init	Transfer	Parameters	IMAGENET Top5
RETINA	Resnet-50	$96.4\% \pm 0.05$	$96.7\% \pm 0.04$	23570408	$92.\% \pm 0.06$
RETINA	Inception-v3	$96.6\% \pm 0.13$	$96.7\% \pm 0.05$	22881424	93.9%
RETINA	CBR-LargeT	$96.2\% \pm 0.04$	$96.2\% \pm 0.04$	8532480	$77.5\% \pm 0.03$
RETINA	CBR-LargeW	$95.8\% \pm 0.04$	$95.8\% \pm 0.05$	8432128	$75.1\% \pm 0.3$
RETINA	CBR-Small	$95.7\% \pm 0.04$	$95.8\% \pm 0.01$	2108672	$67.6\% \pm 0.3$
RETINA	CBR-Tiny	$95.8\% \pm 0.03$	$95.8\% \pm 0.01$	1076480	$73.5\% \pm 0.05$

Table 1: **Transfer learning and random initialization perform comparably across both standard IMAGENET architectures and simple, lightweight CNNs for AUCs from diagnosing moderate DR. Both sets of models also have similar AUCs, despite significant differences in size and complexity.** Model performance on DR diagnosis is also not closely correlated with IMAGENET performance, with the small models performing poorly on IMAGENET but very comparably on the medical task.

Raghu, Maithra, et al. "Transfusion: Understanding transfer learning for medical imaging." Advances in neural information processing systems 32 (2019).

Potential Problems?

- Classify pathologies in chest x-rays

Model Architecture	Atelectasis	Cardiomegaly	Consolidation	Edema	Pleural Effusion
Resnet-50	79.52±0.31	75.23±0.35	85.49±1.32	88.34±1.17	88.70±0.13
Resnet-50 (trans)	79.76±0.47	74.93±1.41	84.42±0.65	88.89±1.66	88.07±1.23
CBR-LargeT	81.52±0.25	74.83±1.66	88.12±0.25	87.97±1.40	88.37±0.01
CBR-LargeT (trans)	80.89±1.68	76.84±0.87	86.15±0.71	89.03±0.74	88.44±0.84
CBR-LargeW	79.79±0.79	74.63±0.69	86.71±1.45	84.80±0.77	86.53±0.54
CBR-LargeW (trans)	80.70±0.31	77.23±0.84	86.87±0.33	89.57±0.34	87.29±0.69
CBR-Small	80.43±0.72	74.36±1.06	88.07±0.60	86.20±1.35	86.14±1.78
CBR-Small (trans)	80.18±0.85	75.24±1.43	86.48±1.13	89.09±1.04	87.88±1.01
CBR-Tiny	80.81±0.55	75.17±0.73	85.31±0.82	84.87±1.13	85.56±0.89
CBR-Tiny (trans)	80.02±1.06	75.74±0.71	84.28±0.82	89.81±1.08	87.69±0.75

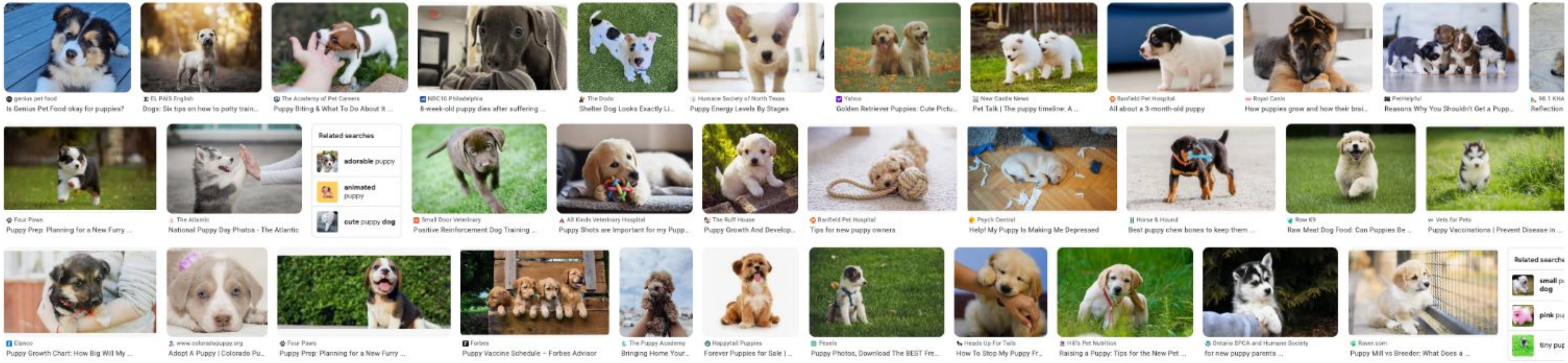
Table 2: **Transfer learning provides mixed performance gains on chest x-rays.** Performances (AUC%) of diagnosing different pathologies on the CHEXPERT dataset. Again we see that transfer learning does not help significantly, and much smaller models performing comparably.

Raghu, Maithra, et al. "Transfusion: Understanding transfer learning for medical imaging." Advances in neural information processing systems 32 (2019).

Cornell Bowers C·IS

Not all images are labeled

- Particular problem for specialized domains (e.g. medicine)
  - Annotation is expensive!
- Much easier to collect unlabeled data
  - Similar to text!
- Can we still learn good image representations?



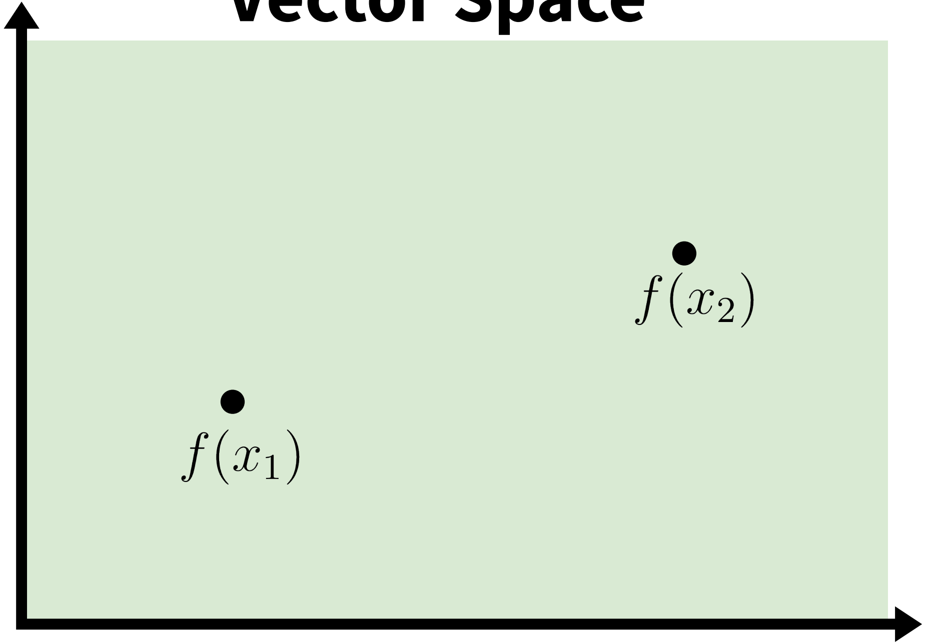
$f(x)$ = raw pixel distance

The exact same image, rotated, maps to a completely different location in vector space

x_1

x_2

Vector Space



Cornell Bowers C·IS

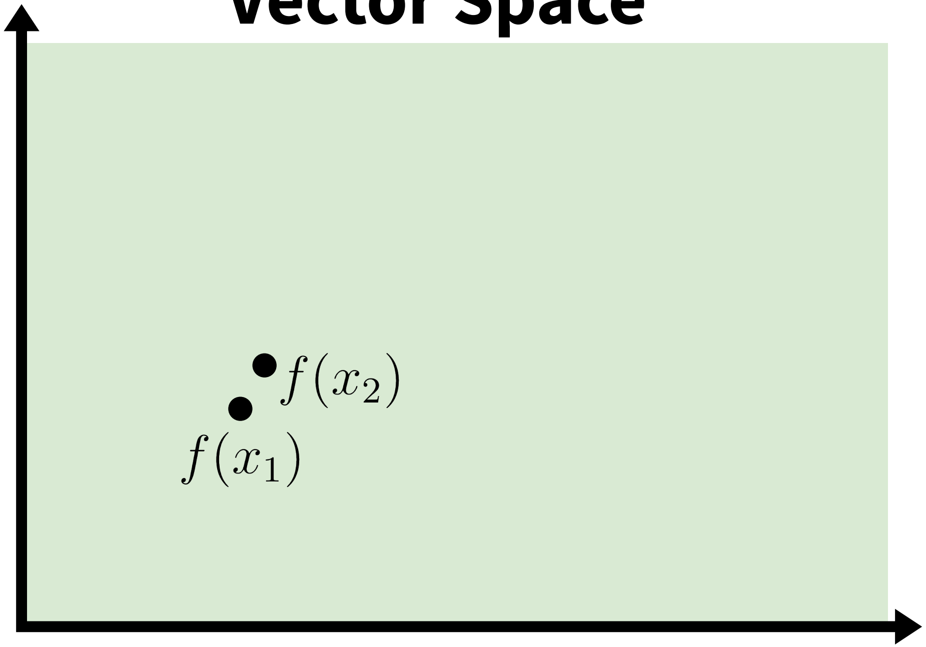
How do we learn structure so that these map to similar points in vector space?

x_1

x_2

$f(x)$ = ???

Vector Space



Cornell Bowers CIS

$f(x)$ = classification network

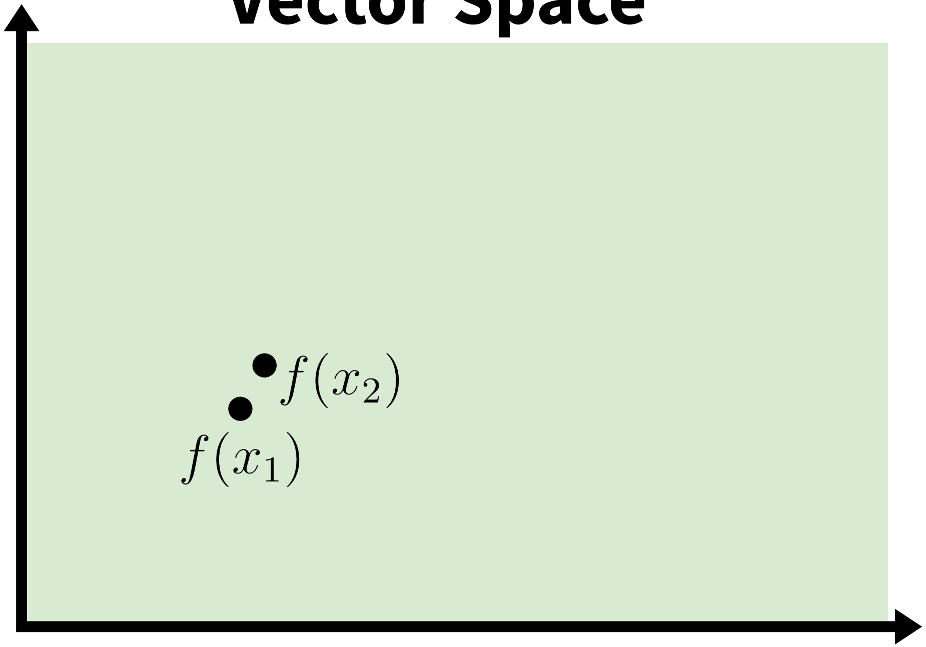
Class "puppy"

Class "puppy"

x_1

x_2

Vector Space

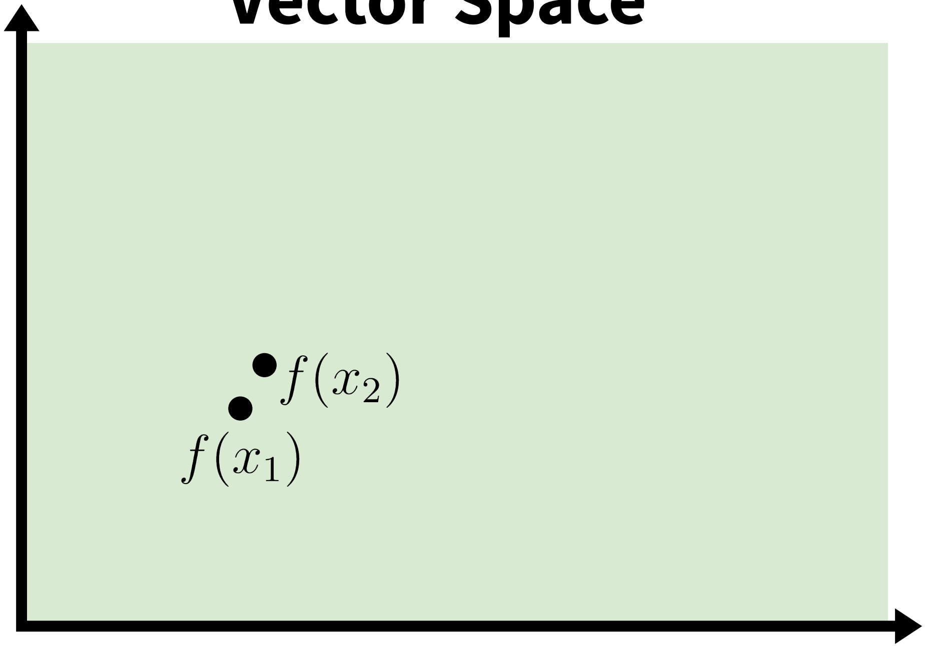


Cornell Bowers C·IS

$f(x)$ = classification network

And what if they are unlabeled?

Vector Space



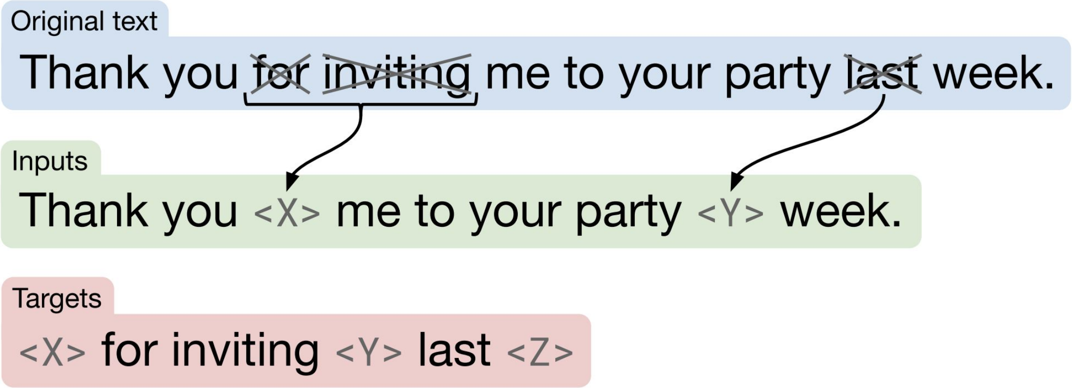
x_1

x_2

Cornell Bowers C·IS

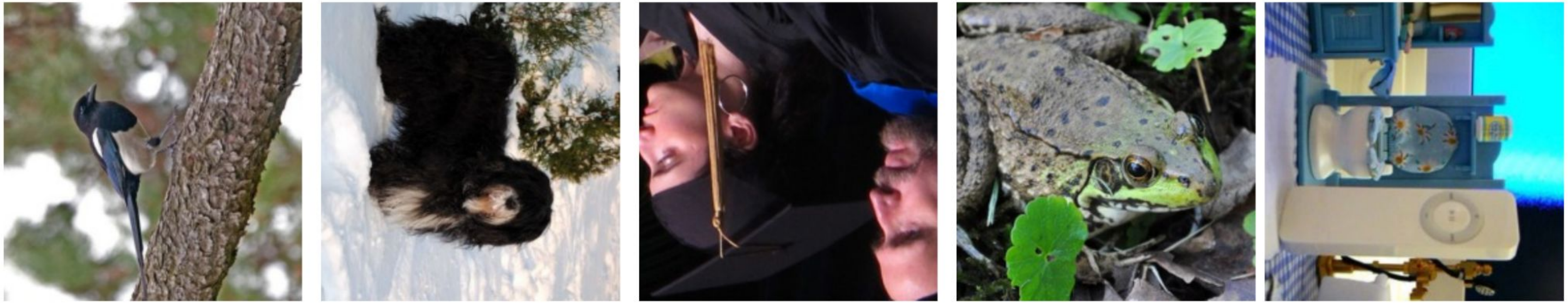
Self-Supervised Learning

- Aim to learn from data without manual label annotation
  - Useful for specialized domains (e.g. medicine) with limited annotated data
- Self-supervised learning methods solve “pretext” tasks that produce good features for downstream tasks.
  - Learn with supervised learning objectives (e.g.,classification, regression)
  - Labels of these pretext tasks are generated automatically



Raffel, Colin, et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." The Journal of Machine Learning Research 21.1 (2020): 5485-5551.

Self-Supervised Learning: Rotation Prediction



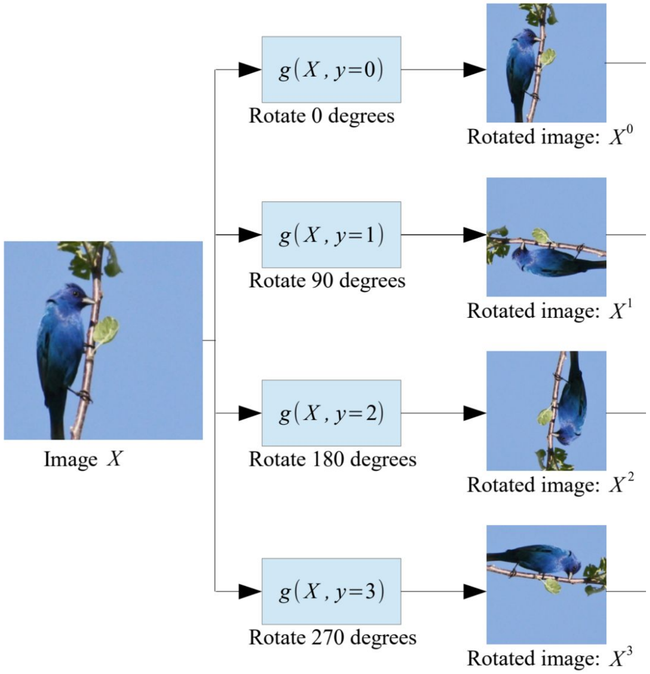
90° rotation | 270° rotation | 180° rotation | 0° rotation | 270° rotation

Figure 1: Images rotated by random multiples of 90 degrees (e.g., 0, 90, 180, or 270 degrees). The core intuition of our self-supervised feature learning approach is that if someone is not aware of the concepts of the objects depicted in the images, he cannot recognize the rotation that was applied to them.

Cornell Bowers C·IS

Rotation Prediction

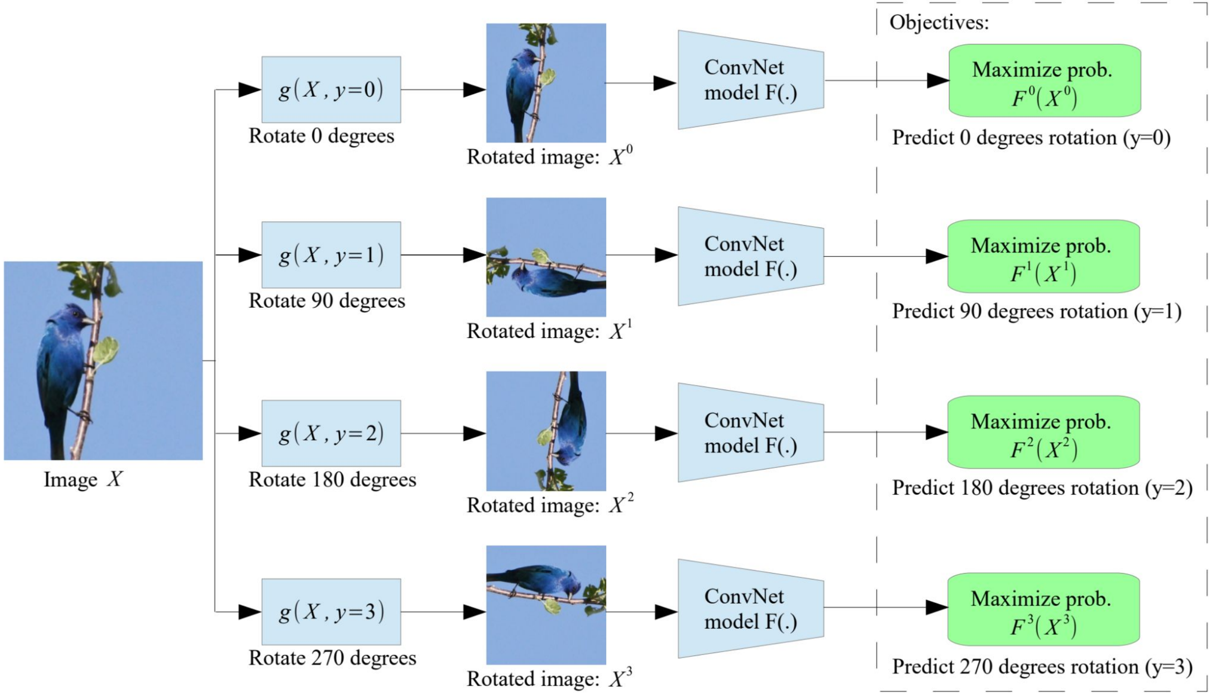
- Self-supervised learning by rotating the input image
- Predict which rotation is applied
  - 4-way classification



Gidaris, Spyros, Praveer Singh, and Nikos Komodakis. "Unsupervised Representation Learning by Predicting Image Rotations." ICLR 2018. 2018.

Rotation Prediction

- Self-supervised learning by rotating the input image
- Predict which rotation is applied
  - 4-way classification

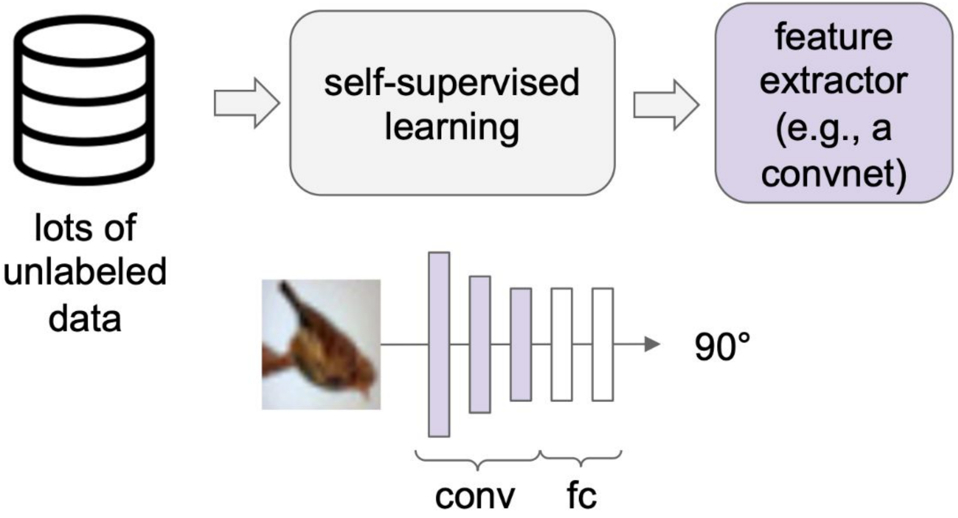


Gidaris, Spyros, Praveer Singh, and Nikos Komodakis. "Unsupervised Representation Learning by Predicting Image Rotations." ICLR 2018. 2018.

How to evaluate a self-supervised learning method?

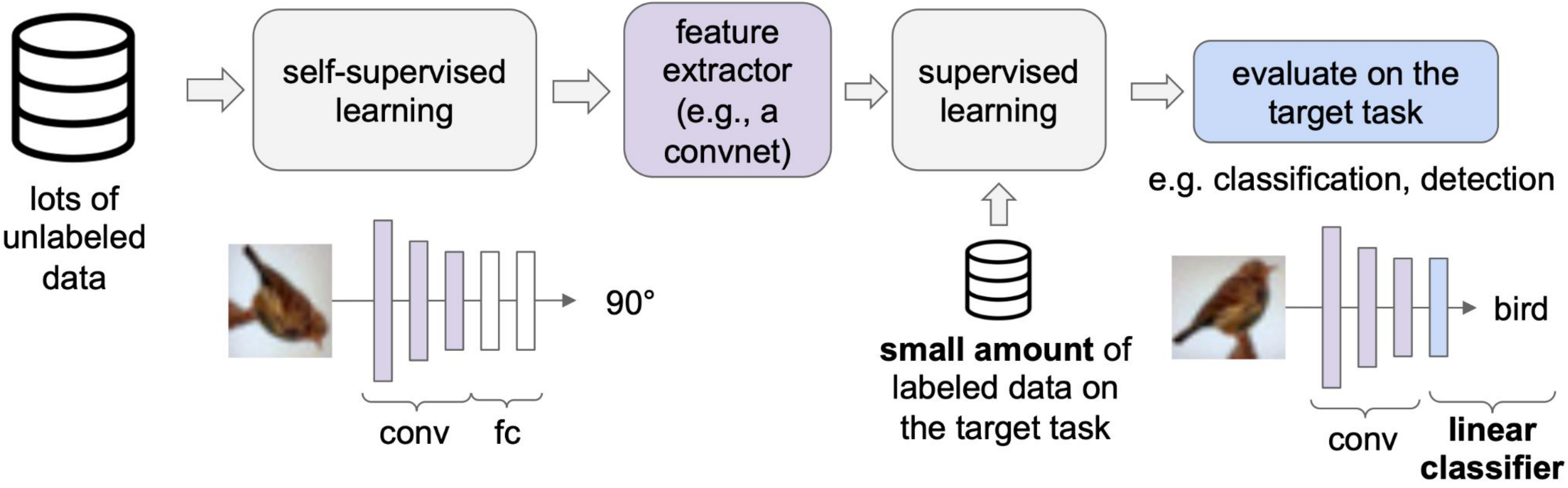
- Don’t care about the performance of the self-supervised learning task
  - E.g. Image rotation prediction
- Evaluate the learned feature encoder on downstream target tasks

How to evaluate a self-supervised learning method?



1. Learn good feature extractors from self-supervised pretext tasks, e.g., predicting image rotations

How to evaluate a self-supervised learning method?



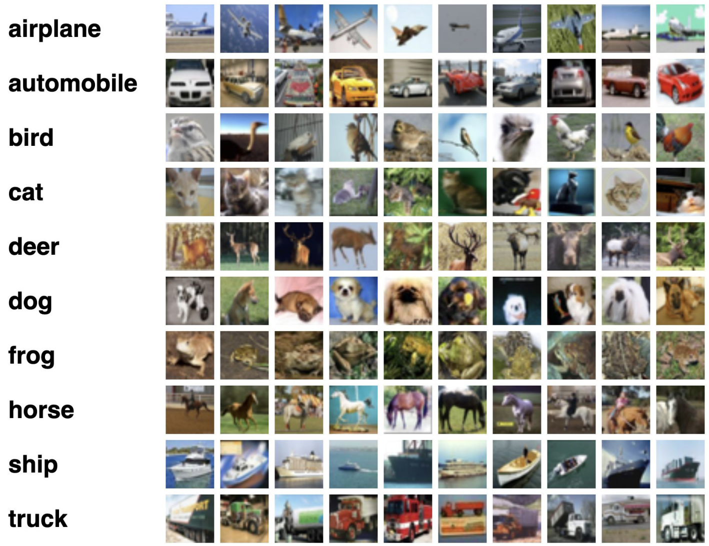
1. Learn good feature extractors from self-supervised pretext tasks, e.g., predicting image rotations

2. Attach a shallow network on the feature extractor; train the shallow network on the target task with small amount of labeled data

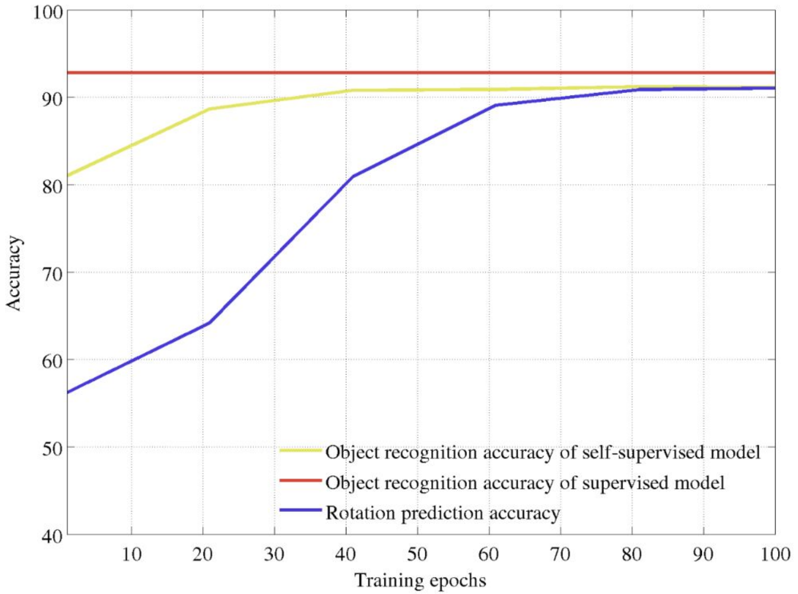
Cornell Bowers C·IS

Self-Supervised Evaluation

- Downstream performance correlates with prefix task: rotation prediction



Cifar-10 Image Classification

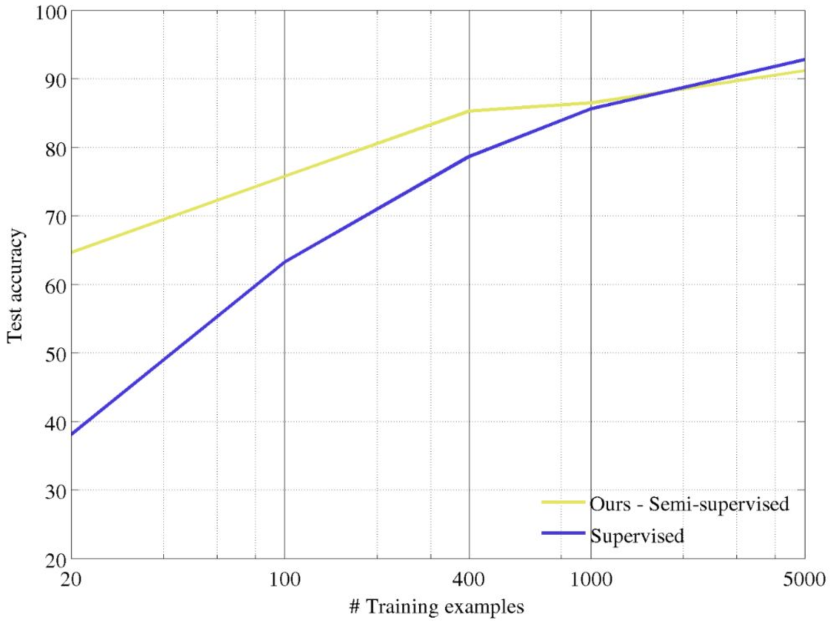


Gidaris, Spyros, Praveer Singh, and Nikos Komodakis. "Unsupervised Representation Learning by Predicting Image Rotations." ICLR 2018. 2018.

Cornell Bowers C·IS

Self-Supervised Evaluation

- Self-supervised learning outperforms supervised learning with limited data
  - Can use large volumes of unlabeled data!



Gidaris, Spyros, Praveer Singh, and Nikos Komodakis. "Unsupervised Representation Learning by Predicting Image Rotations." ICLR 2018. 2018.

Discuss

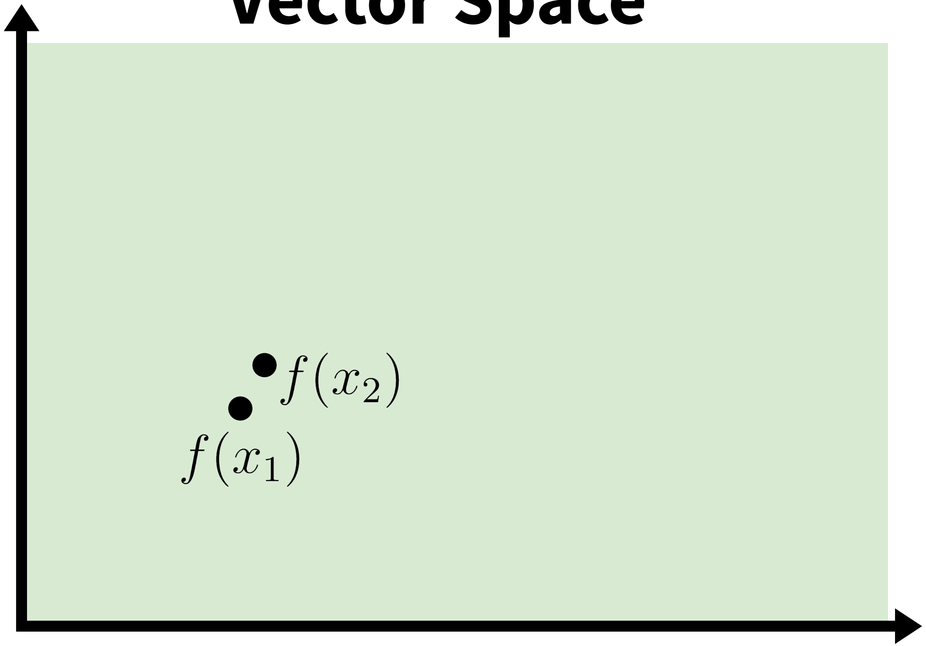
We are provided this image without labels: what are some other tasks we can do with it?

How can we perform self-supervised learning with images?

$$f(x) = ???$$

Can we learn this directly?

Vector Space

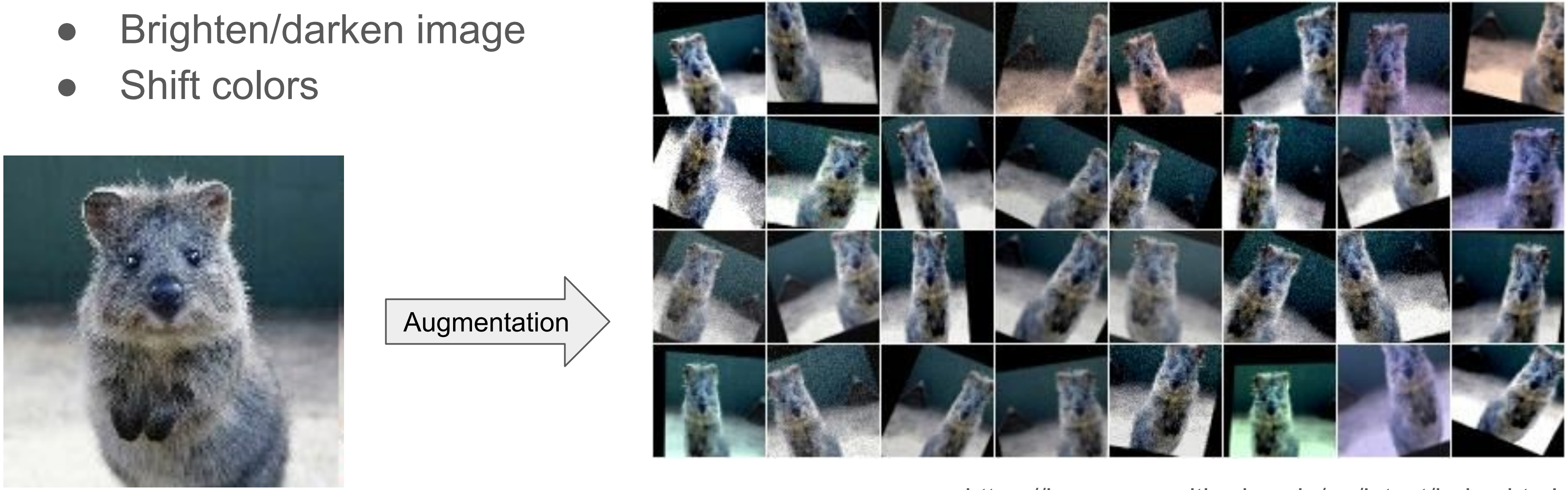


 x_1

 x_2

Review: Image Augmentation

- Horizontal flips
- Rotate image
- Zoom/crop image
- Brighten/darken image
- Shift colors



https://imgaug.readthedocs.io/en/latest/index.html

Cornell Bowers CIS

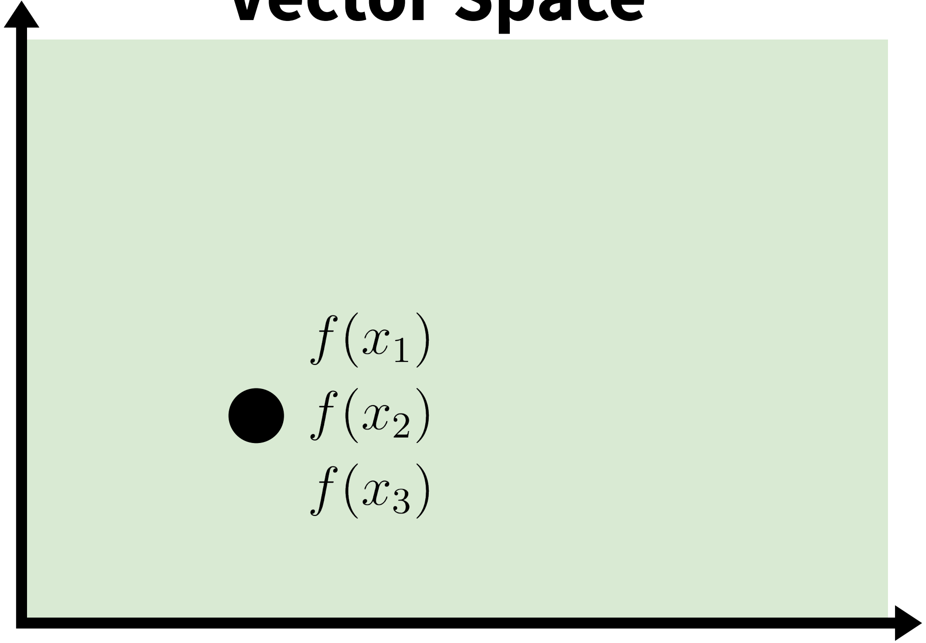
$f(x)$= contrastive learning

x_1

x_2

x_3

Vector Space



All positive pairs are augmentations of the original image

Cornell Bowers C·IS

$f(x)$= contrastive learning

x_4

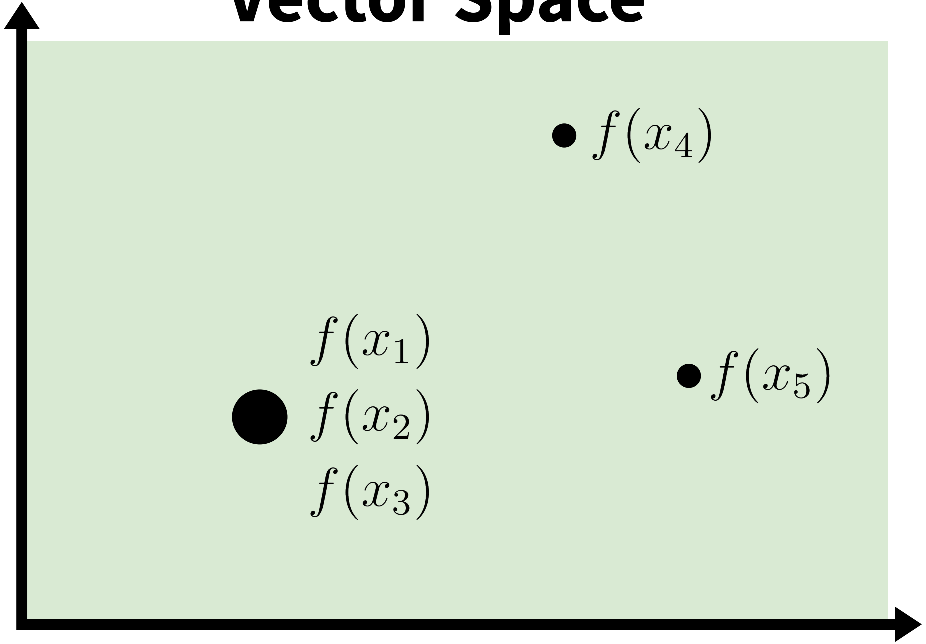
x_5

x_1

x_2

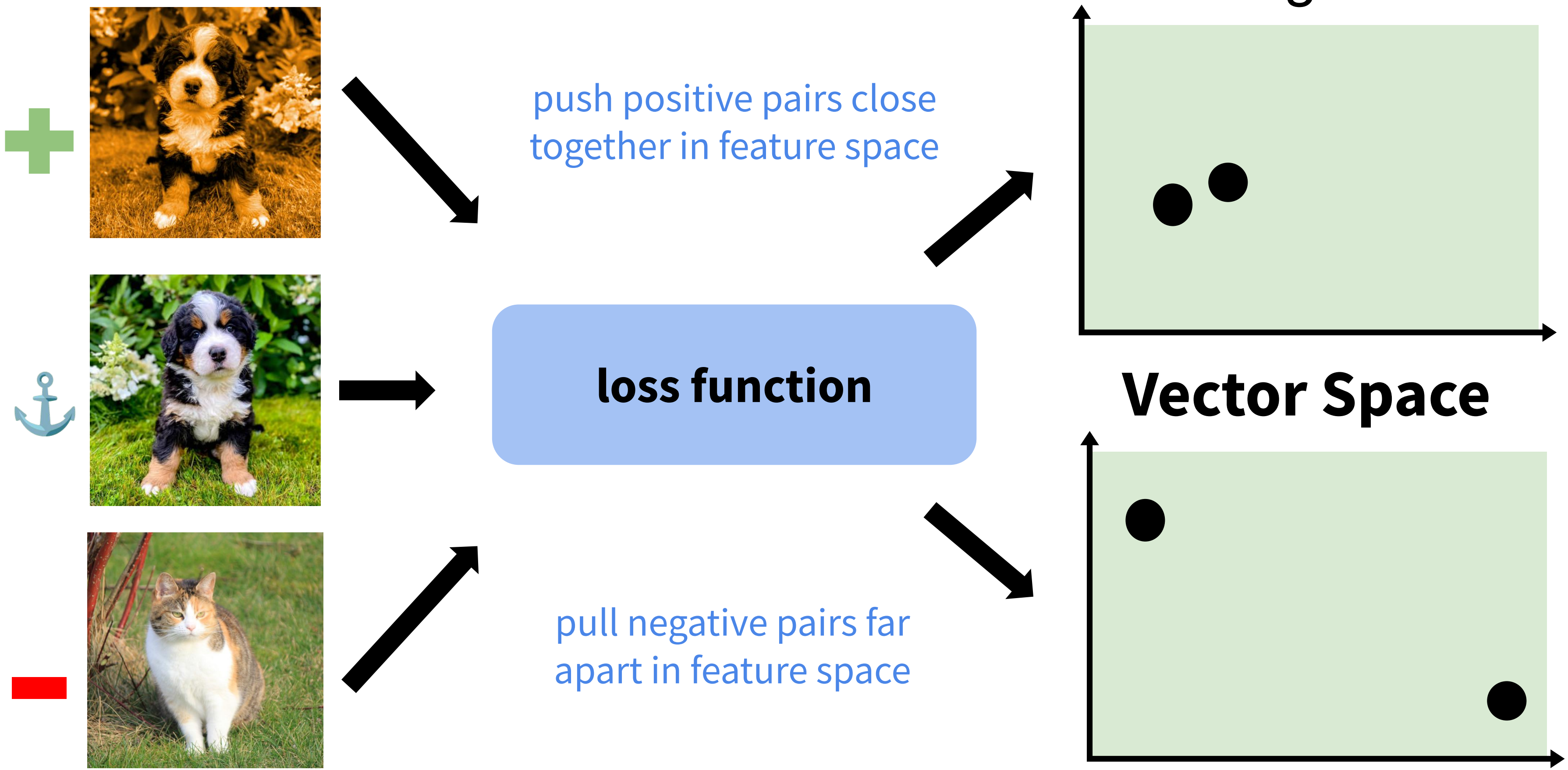
x_3

Vector Space



Any other image is a negative pair

Basic Model for Contrastive Learning



Triplet loss function

Anchor example

$$\ell = \max(0, \|f(\mathbf{x_i}) - f(\mathbf{x}^+)\|^2 - \|f(\mathbf{x_i}) - f(\mathbf{x}^-)\|^2 + c)$$

Model

Positive pair

Negative pair

Triplet loss function

Margin

$$\ell = \max(0, \underbrace{\|f(\mathbf{x_i}) - f(\mathbf{x}^+)\|^2}_{\text{Model should map positive examples close together}} - \underbrace{\|f(\mathbf{x_i}) - f(\mathbf{x}^-)\|^2}_{\text{Model should map negative examples far apart}} + c)$$

Ensures loss is not negative

Discuss

- Any potential problems with the triplet loss?
- Any ideas to remedy those problems

Margin

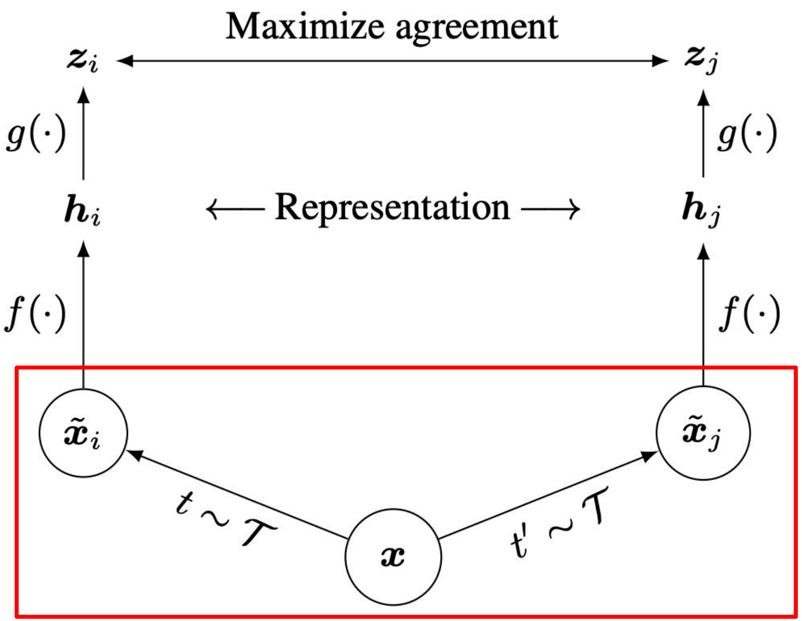
$$\ell = \max(0, \underbrace{\|f(\mathbf{x_i}) - f(\mathbf{x}^+)\|^2}_{\text{Model should map positive examples close together}} - \underbrace{\|f(\mathbf{x_i}) - f(\mathbf{x}^-)\|^2}_{\text{Model should map negative examples far apart}} + c)$$

Ensures loss is not negative

Cornell Bowers C·IS

SimCLR: A Simple Contrastive Learning Framework for Images

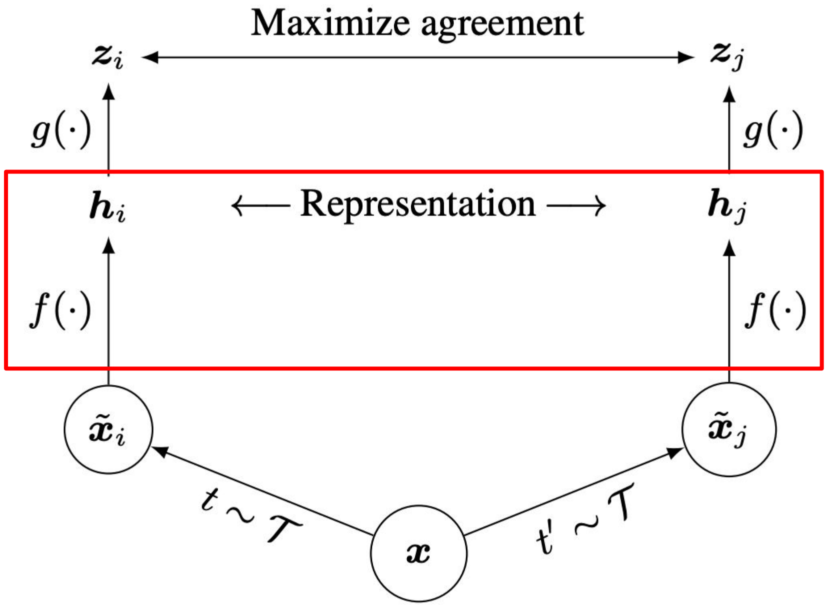
- Sample two different augmentations of an image



Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." International conference on machine learning. PMLR, 2020.

SimCLR: A Simple Contrastive Learning Framework for Images

- Sample two different augmentations of an image
- Apply a base encoder to each view of the image to extract an image feature
  - e.g. ResNet

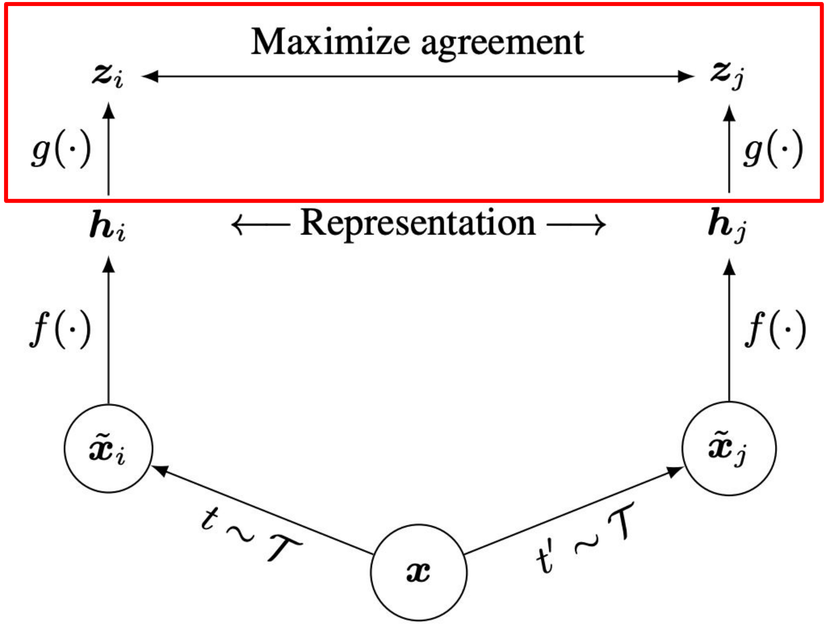


Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." International conference on machine learning. PMLR, 2020.

Cornell Bowers C·IS

SimCLR: A Simple Contrastive Learning Framework for Images

- Sample two different augmentations of an image
- Apply a base encoder to each view of the image to extract an image feature
  - e.g. ResNet
- Apply an MLP projection head to generate final representations
  - Throw away projection head after training

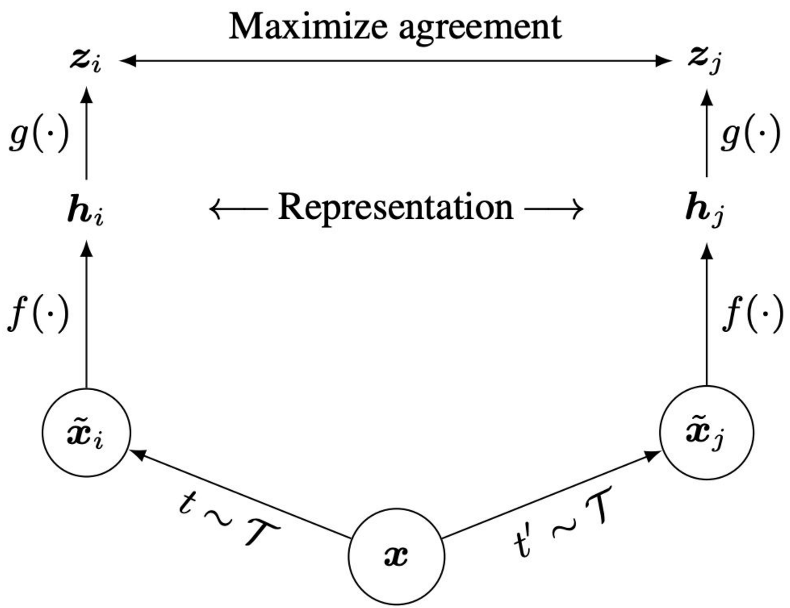


Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." International conference on machine learning. PMLR, 2020.

Cornell Bowers C·IS

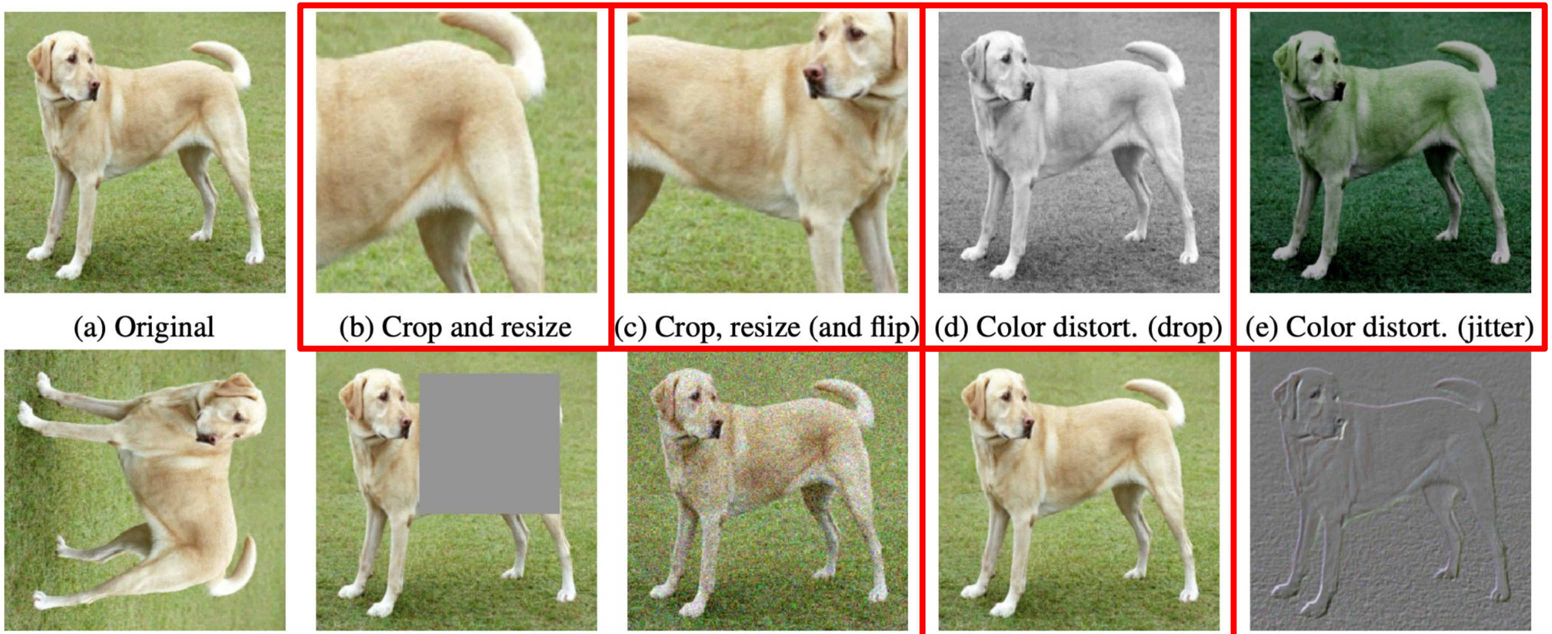
SimCLR: A Simple Contrastive Learning Framework for Images

- Sample two different augmentations of an image
- Apply a base encoder to each view of the image to extract an image feature
  - e.g. ResNet
- Apply an MLP projection head to generate final representations
  - Throw away projection head after training



Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." International conference on machine learning. PMLR, 2020.

SimCLR Augmentations



(a) Original (b) Crop and resize (c) Crop, resize (and flip) (d) Color distort. (drop) (e) Color distort. (jitter)

(f) Rotate $\{90°, 180°, 270°\}$ (g) Cutout (h) Gaussian noise (i) Gaussian blur (j) Sobel filtering

Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." International conference on machine learning. PMLR, 2020.

SimCLR Loss

- Temperature-scaled cross-entropy loss

$$\mathcal{L}_{\text{SimCLR}} = -\log\left(\frac{\overbrace{\exp(d(\mathbf{x}_i,\ \mathbf{x}_i^+)/\tau)}^{\text{Model should map positive examples close together}}}{\exp(d(\mathbf{x}_i,\ \mathbf{x}_i^+)/\tau) + \underbrace{\exp(d(\mathbf{x}_i,\ \mathbf{x}_i^-)/\tau)}_{\text{Model should map negative examples far apart}}}\right)$$

Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." International conference on machine learning. PMLR, 2020.

SimCLR Algorithm

- Use other images in the mini-batch as negatives
- L2 normalize representations
  - Use cosine similarity as the distance metric
- Compute temperature-scaled cross-entropy for all positive pairs

Algorithm 1 SimCLR's main learning algorithm.

input: batch size N, constant τ, structure of f, g, $\mathcal{T}$.
for sampled minibatch $\{\boldsymbol{x}_k\}_{k=1}^N$ **do**
  for all $k \in \{1, \dots, N\}$ **do**
    draw two augmentation functions $t \sim \mathcal{T}$, $t' \sim \mathcal{T}$
    # the first augmentation
    $\tilde{\boldsymbol{x}}_{2k-1} = t(\boldsymbol{x}_k)$
    $\boldsymbol{h}_{2k-1} = f(\tilde{\boldsymbol{x}}_{2k-1})$ # representation
    $\boldsymbol{z}_{2k-1} = g(\boldsymbol{h}_{2k-1})$ # projection
    # the second augmentation
    $\tilde{\boldsymbol{x}}_{2k} = t'(\boldsymbol{x}_k)$
    $\boldsymbol{h}_{2k} = f(\tilde{\boldsymbol{x}}_{2k})$ # representation
    $\boldsymbol{z}_{2k} = g(\boldsymbol{h}_{2k})$ # projection
  end for
  for all $i \in \{1, \dots, 2N\}$ and $j \in \{1, \dots, 2N\}$ **do**
    $s_{i,j} = \boldsymbol{z}_i^\top \boldsymbol{z}_j / (\|\boldsymbol{z}_i\| \|\boldsymbol{z}_j\|)$ # pairwise similarity
  end for
  define $\ell(i,j)$ **as** $\ell(i,j) = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(s_{i,k}/\tau)}$
  $\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} [\ell(2k-1, 2k) + \ell(2k, 2k-1)]$
  update networks f and g to minimize $\mathcal{L}$
end for
return encoder network $f(\cdot)$, and throw away $g(\cdot)$

Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." International conference on machine learning. PMLR, 2020.

Cornell Bowers C·IS

Comparison of Loss Functions

- Temperature-scaled cross entropy places more weight on hard negatives
  - Don’t need to mine hard negatives

Name	Negative loss function	Gradient w.r.t. $\boldsymbol{u}$
NT-Xent	$\boldsymbol{u}^T\boldsymbol{v}^+/\tau - \log\sum_{\boldsymbol{v}\in\{\boldsymbol{v}^+,\boldsymbol{v}^-\}}\exp(\boldsymbol{u}^T\boldsymbol{v}/\tau)$	$(1-\frac{\exp(\boldsymbol{u}^T\boldsymbol{v}^+/\tau)}{Z(\boldsymbol{u})})/\tau\boldsymbol{v}^+ - \sum_{\boldsymbol{v}^-}\frac{\exp(\boldsymbol{u}^T\boldsymbol{v}^-/\tau)}{Z(\boldsymbol{u})}/\tau\boldsymbol{v}^-$
NT-Logistic	$\log\sigma(\boldsymbol{u}^T\boldsymbol{v}^+/\tau) + \log\sigma(-\boldsymbol{u}^T\boldsymbol{v}^-/\tau)$	$(\sigma(-\boldsymbol{u}^T\boldsymbol{v}^+/\tau))/\tau\boldsymbol{v}^+ - \sigma(\boldsymbol{u}^T\boldsymbol{v}^-/\tau)/\tau\boldsymbol{v}^-$
Margin Triplet	$-\max(\boldsymbol{u}^T\boldsymbol{v}^- - \boldsymbol{u}^T\boldsymbol{v}^+ + m, 0)$	$\boldsymbol{v}^+ - \boldsymbol{v}^-$ if $\boldsymbol{u}^T\boldsymbol{v}^+ - \boldsymbol{u}^T\boldsymbol{v}^- < m$ else $\mathbf{0}$

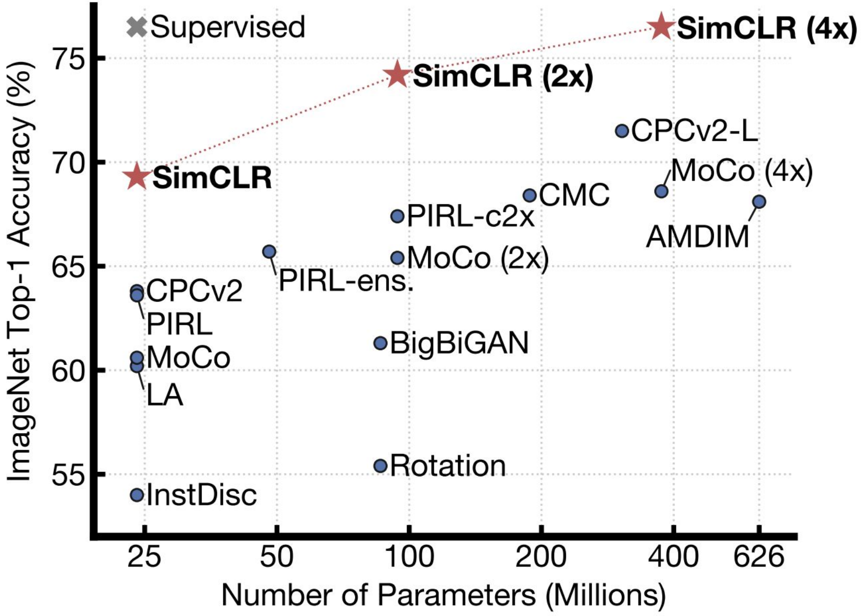
Table 2. Negative loss functions and their gradients. All input vectors, i.e. $\boldsymbol{u}, \boldsymbol{v}^+, \boldsymbol{v}^-$, are ℓ_2 normalized. NT-Xent is an abbreviation for “Normalized Temperature-scaled Cross Entropy”. Different loss functions impose different weightings of positive and negative examples.

Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." International conference on machine learning. PMLR, 2020.

Cornell Bowers C·IS

SimCLR Results

- Train a linear classifier on features from SimCLR
- Approaches supervised performance!



Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." International conference on machine learning. PMLR, 2020.

SimCLR Results

- Self-supervised vs. supervised ImageNet pre-training
- Evaluate transfer performance across 12 downstream classification datasets
  - Often outperforms supervised pre-training!

	Food	CIFAR10	CIFAR100	Birdsnap	SUN397	Cars	Aircraft	VOC2007	DTD	Pets	Caltech-101	Flowers
Linear evaluation:												
SimCLR (ours)	**76.9**	**95.3**	80.2	48.4	**65.9**	60.0	61.2	**84.2**	**78.9**	89.2	**93.9**	**95.0**
Supervised	75.2	**95.7**	**81.2**	**56.4**	64.9	**68.8**	**63.8**	83.8	**78.7**	**92.3**	**94.1**	94.2
Fine-tuned:												
SimCLR (ours)	**89.4**	**98.6**	**89.0**	**78.2**	**68.1**	**92.1**	**87.0**	**86.6**	**77.8**	92.1	**94.1**	97.6
Supervised	88.7	98.3	**88.7**	**77.8**	67.0	91.4	**88.0**	86.5	**78.8**	**93.2**	**94.2**	**98.0**
Random init	88.3	96.0	81.9	**77.0**	53.7	91.3	84.8	69.4	64.1	82.7	72.5	92.5

Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." International conference on machine learning. PMLR, 2020.

Cornell Bowers C·IS

Effect Of Projection Head

- Projects data to "augmentation-invariant" representation
  - Less useful features for downstream tasks

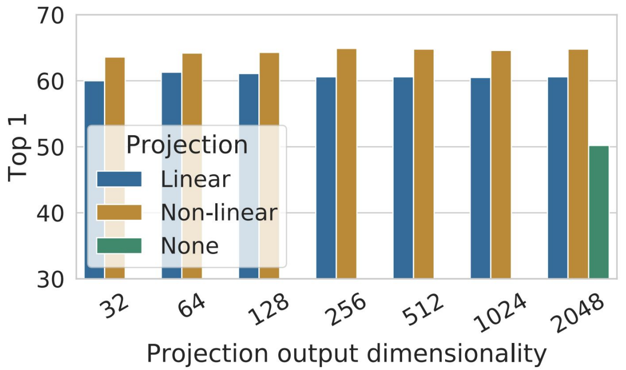
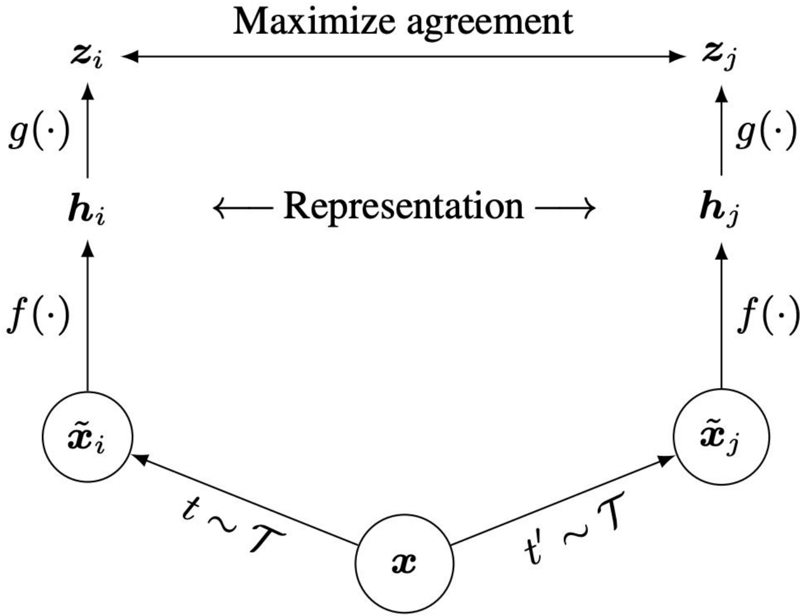


Figure 8. Linear evaluation of representations with different projection heads $g(\cdot)$ and various dimensions of $\boldsymbol{z} = g(\boldsymbol{h})$. The representation $\boldsymbol{h}$ (before projection) is 2048-dimensional here.

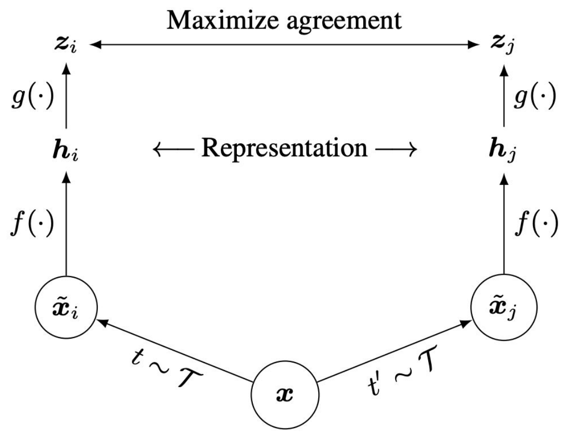


Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." International conference on machine learning. PMLR, 2020.

Cornell Bowers CIS

Effect Of Projection Head

- Projects data to “augmentation-invariant” representation
  - Features less useful for downstream tasks



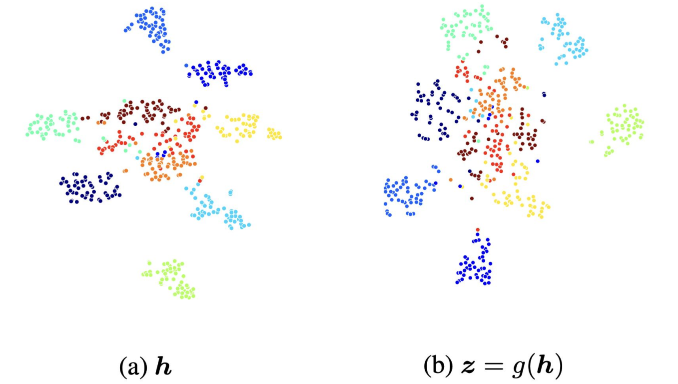


Figure B.4. t-SNE visualizations of hidden vectors of images from a randomly selected 10 classes in the validation set.

What to predict?	Random guess	Representation $\boldsymbol{h}$	$g(\boldsymbol{h})$
Color vs grayscale	80	99.3	97.4
Rotation	25	67.6	25.6
Orig. vs corrupted	50	99.5	59.6
Orig. vs Sobel filtered	50	96.6	56.3

Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." International conference on machine learning. PMLR, 2020.

Impact of Loss Function

- Proposed loss outperforms the margin loss
    - Even with negative mining
- L2 normalization is useful
- Sensitive to cross-entropy temperature

Margin	NT-Logi.	Margin (sh)	NT-Logi.(sh)	NT-Xent
50.9	51.6	57.5	57.9	63.9

Table 4. Linear evaluation (top-1) for models trained with different loss functions. "sh" means using semi-hard negative mining.

ℓ_2 norm?	τ	Entropy	Contrastive acc.	Top 1
Yes	0.05	1.0	90.5	59.7
	0.1	4.5	87.8	64.4
	0.5	8.2	68.2	60.7
	1	8.3	59.1	58.0
No	10	0.5	91.7	57.2
	100	0.5	92.1	57.0

Table 5. Linear evaluation for models trained with different choices of ℓ_2 norm and temperature τ for NT-Xent loss. The contrastive distribution is over 4096 examples.

Impact of Batch Size

- Requires large batches
  - Harder negatives!

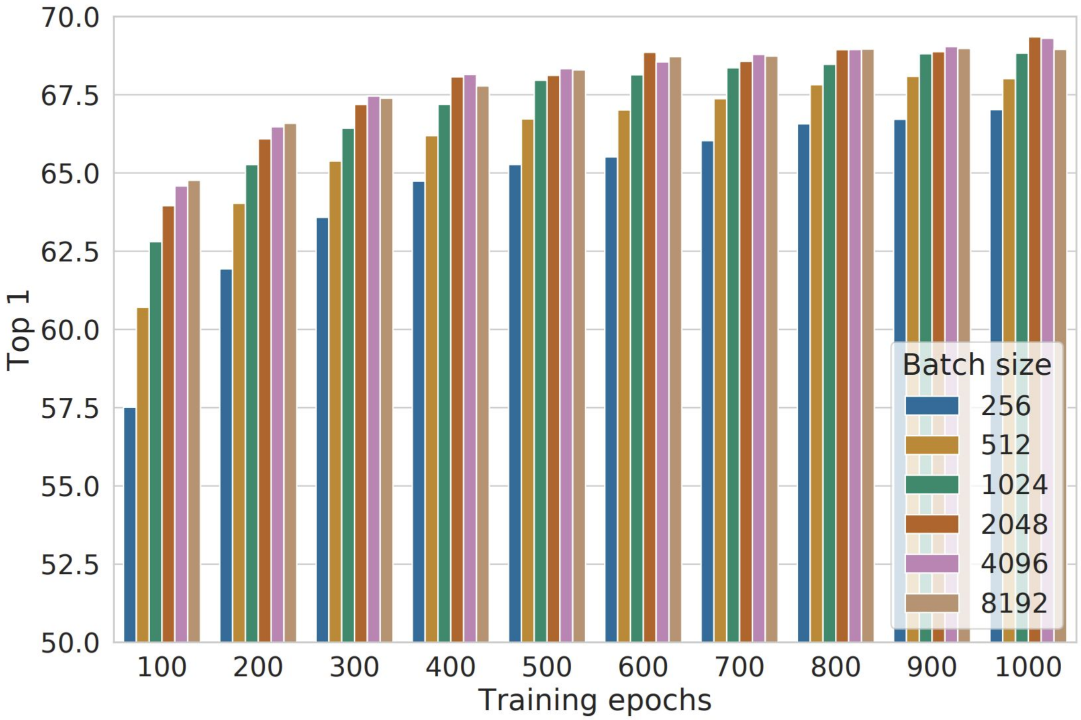
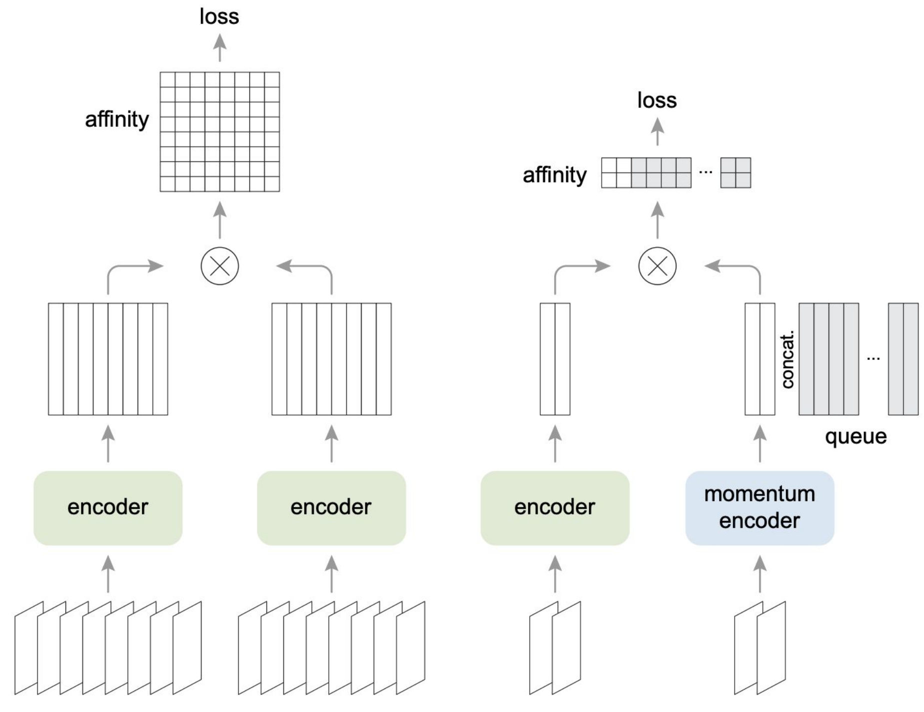


Figure 9. Linear evaluation models (ResNet-50) trained with different batch size and epochs. Each bar is a single run from scratch.[10]

Momentum Contrast (MoCo)

- Cache negative samples from earlier batches as you train
- Replace one encoder with an exponential moving average (EMA) of the model
  - Makes queued representations more stable

$$\theta_{\mathrm{k}} \leftarrow m\theta_{\mathrm{k}} + (1-m)\theta_{\mathrm{q}}$$



(a) end-to-end (b) Momentum Contrast

Chen, Xinlei, et al. "Improved baselines with momentum contrastive learning." arXiv preprint arXiv:2003.04297 (2020).

Cornell Bowers CIS

MoCo v2

- MoCo v2: MoCo with some tricks from SimCLR
  - Stronger augmentations
  - MLP projection head
- Outperform SimCLR with modest batch sizes
  - Large numbers of negatives available from the queue

case	unsup. pre-train MLP	aug+	cos	epochs	batch	ImageNet acc.
MoCo v1 [6]				200	256	60.6
SimCLR [2]	✓	✓	✓	200	256	61.9
SimCLR [2]	✓	✓	✓	200	8192	66.6
MoCo v2	✓	✓	✓	200	256	**67.5**
*results of **longer** unsupervised training follow:*						
SimCLR [2]	✓	✓	✓	1000	4096	69.3
MoCo v2	✓	✓	✓	800	256	**71.1**

Table 2. **MoCo *vs.* SimCLR**: ImageNet linear classifier accuracy (**ResNet-50, 1-crop 224×224**), trained on features from unsupervised pre-training. "aug+" in SimCLR includes blur and stronger color distortion. SimCLR ablations are from Fig. 9 in [2] (we thank the authors for providing the numerical results).

mechanism	batch	memory / GPU	time / 200-ep.
MoCo	256	**5.0G**	**53 hrs**
end-to-end	256	7.4G	65 hrs
end-to-end	4096	93.0G†	n/a

Table 3. **Memory and time cost** in 8 V100 16G GPUs, implemented in PyTorch. †: based on our estimation.

Chen, Xinlei, et al. "Improved baselines with momentum contrastive learning." arXiv preprint arXiv:2003.04297 (2020).

Cornell Bowers C·IS

Recap

- Supervised image classification pre-training produces strong image representations
  - Can efficiently transfer to other tasks
- Can apply self-supervised learning to images
  - Prefix tasks: rotation prediction, masked-image modeling, etc.
- Contrastive learning explicitly enforces similarity in representation space
  - Requires defining image augmentations