Unsupervised Pre-Training: Contrastive Learning

CS 330

Course Reminders

Project proposal due Wednesday.

(graded lightly, for your benefit)

Homework 2 due next Monday 10/24.

Following up on some high-res feedback:

- I will work on making whiteboard writing larger.
- Moving one TA office hours (Garrett) from in-person-> over zoom.
- Will clarify project expectations on Ed.

So Far

Few-shot learning via meta-learning

Problem: Given data from $\mathcal{T}_1, \ldots, \mathcal{T}_n$, solve new task $\mathcal{T}_{\text{test}}$ more quickly / proficiently / stably

Methods: black-box, optimization-based, non-parametric

What if you don't have a lot of tasks?

What if you *only* have **one batch** of **unlabeled data**?

This Lecture

Unsupervised representation learning for few-shot learning

Part I: Contrastive learning

Part II (next time): Reconstruction-based methods

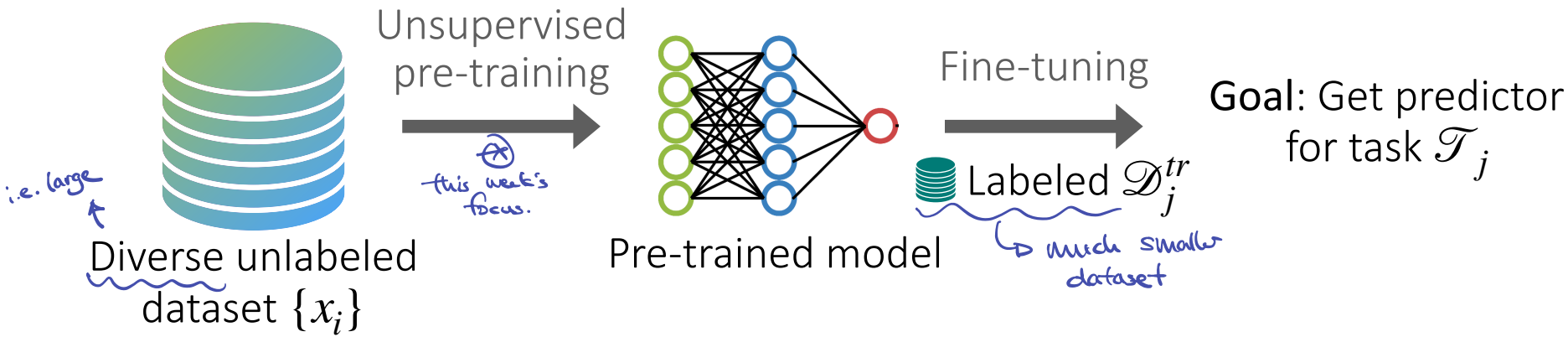
Relation to meta-learning. → pretty closely related!

Goals for the lecture:

- Understand **contrastive learning**: intuition, design choices, how to implement
- How contrastive learning relates to meta-learning

Unsupervised Pre-Training Set-Up

↳ Did this for point cloud data + SimCLR.

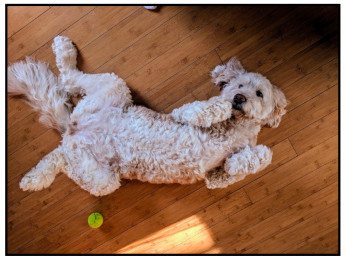


Key Idea of Contrastive Learning

Similar examples should have **similar representations**

Key idea of contrastive learning!

Examples with the same class label

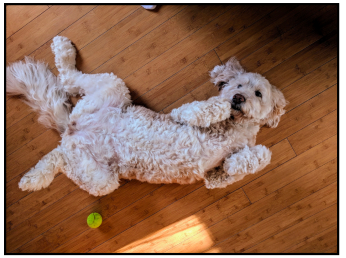



(Requires labels, related to Siamese nets, ProtoNets)

↳ Covered last class

Augmented versions of the example

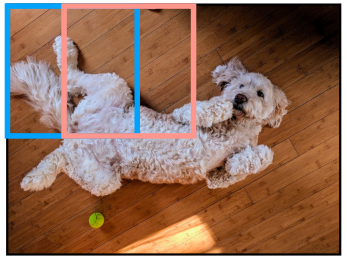
SimCLR paper




(flip & crop)

Diff types of similar images

Nearby image patches



Nearby video frames


van den Oord, Li, Vinyals. CPC. 2018

Chen, Kornblith, Norouzi, Hinton. SimCLR. ICML 2020

Dog credit to Maggie & Luke

Key Idea of Contrastive Learning

Similar examples should have **similar representations**

Similar representations

Similar representations

Important

Question: Why not simply minimize difference between representations?

$$\min_{\theta} \sum_{(x_i,x_j)} \|f_\theta(x_i) - f_\theta(x_j)\|^2$$

Need to both **compare** & *contrast*!

→ Converges to a degenerate solution where all images get mapped to a single constant vector, i.e. $\mathcal{L} \to 0$

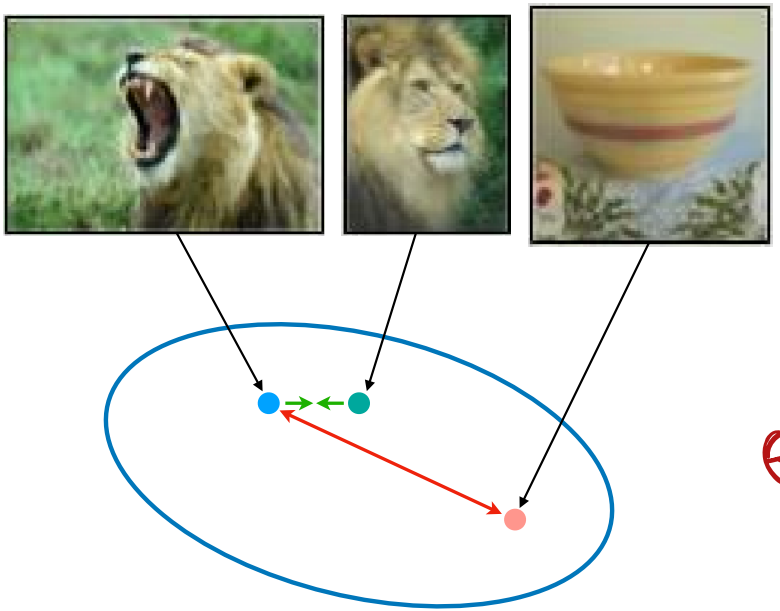
⇒ Don't get a good representation... just collapses.

⇒ Hence the "Contrastive" learning

Key Idea of Contrastive Learning

Similar examples should have **similar representations**

Need to both **compare** & *contrast*!



Embedding space $f_\theta(x)$

Bring together representations of similar examples.

Push apart representations of differing examples.

"Contrastive"

Key design choices:

1. Implementation of contrastive loss
2. Choosing what to compare/contrast

이거 원안이랑

Contrastive Learning Implementation

Hinge Loss (applied in the eqn.)

Similar examples should have **similar representations**

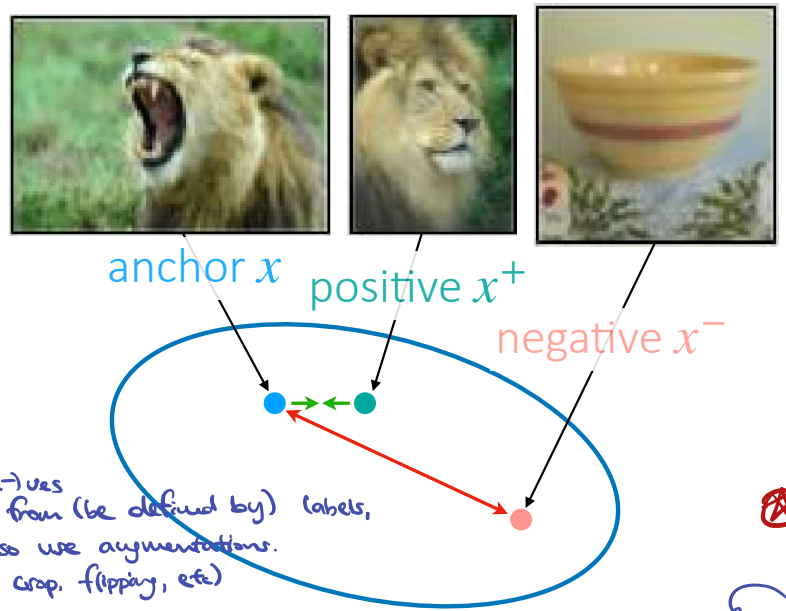
$\mathcal{L}(y) = \max(0, 1 - \epsilon \cdot y)$

Need to both compare & *contrast*!

Q: Is there a way to push some examples more than others?

A. In general CC pushes away the same for all (-)ves. Examples may end up in diff. distances ∴ some will be naturally be harder to push away.

eg) wolf, wolf dog, dog ⟷ cat



(+)ve & (-)ves can come from (be defined by) labels, or! can also use augmentations. (random crop, flipping, etc)

not always exactly class labels

Simplest form

V1. Triplet loss:

anchor positive anchor negative

$$\min_\theta \sum_{(x,x^+,x^-)} \max\left(0, \|f_\theta(x) - f_\theta(x^+)\|^2 - \|f_\theta(x) - f_\theta(x^-)\|^2 + \epsilon\right)$$

can use Euclidean or negative cosine similarity

this is unbounded ⇒ will go to (-)ve inf. unless you take this into account when designing embedding func. or the loss function.

Compare to **Siamese networks**:

Classify (x, x') as same class if $\|f(x) - f(x')\|^2$ is small.

Key difference: learns a metric space, not just a classifier

↳ triplet loss

Challenge: need to find difficult negatives.

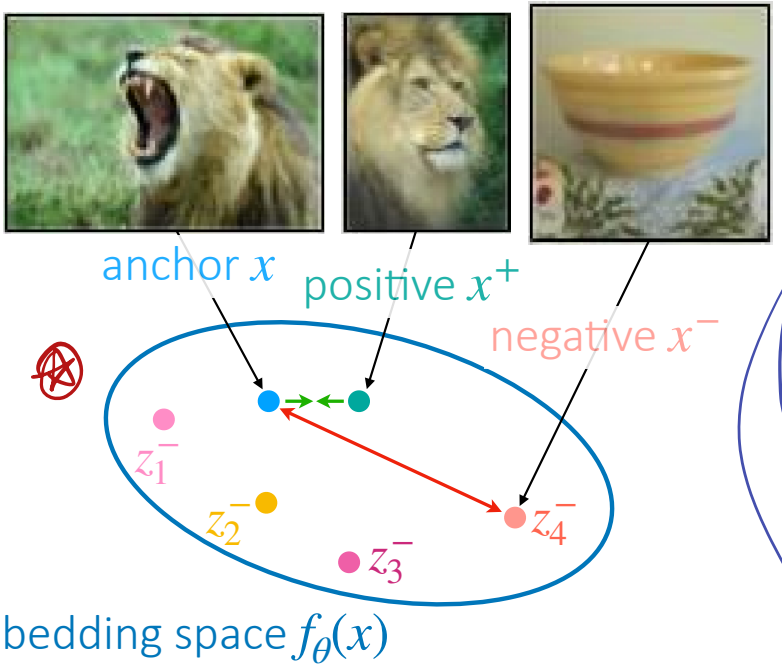
- If you sample a (-)ve that's already v. far away, then you'll just get $\mathcal{L}=0$, & the model will not learn.
- "Hard Negative" helps find (-)ves that allows the model to continually learn.

Schroff, Kalenichenko, Philbin. CVPR 2015

Contrastive Learning Implementation

Similar examples should have **similar representations**

Need to both **compare** & *contrast*!



Trying to classify whether an example is (+)ve or (−)ve, given an anchor (same w/ V1 – triplet loss)

V2. From binary to N-way classification:

↳ rather than binaries ↳ multiple negatives

$$\mathcal{L}_{\text{N-way}}(\theta) = -\sum_{z} \log \frac{\exp(-d(z, z^+))}{\sum_i \exp(-d(z, z_i^-))}$$

↳ Used in SimCLR paper!

$z = f_\theta(x)$

- generalization of triplet loss to multiple negatives

↳ v. similar to summing up the negative loss for triplet loss instead of Σ, it's doing Σ-log.

Softmax $(-d(\text{anchor}, \underline{\text{example}}))$ ↳ could be both (+)ve + (−)ve

Alternatively, can sum all examples in the denominator

Sohn. N-Pair Loss Objective. NIPS 2016

Chen, Kornblith, Norouzi, Hinton. SimCLR. ICML 2020

Contrastive Learning Implementation

Generating good augmentations is key.

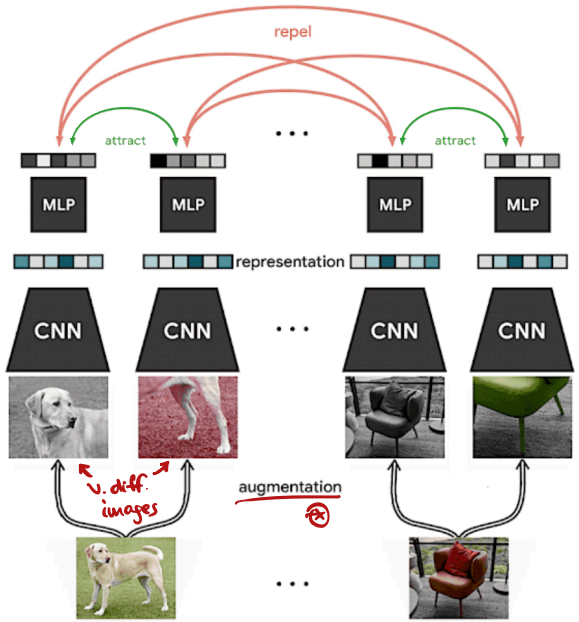
SimCLR Algorithm

Unsupervised Pre-Training

generates more (-)ves
can augment more or less.

1. Sample minibatch of examples $x_1, \ldots, x_N$
2. **Augment** each example twice to get $\tilde{x}_1, \ldots, \tilde{x}_N, \tilde{x}'_1, \ldots, \tilde{x}'_N$
3. **Embed examples** with f_θ to get $\tilde{z}_1, \ldots, \tilde{z}_N, \tilde{z}'_1, \ldots, \tilde{z}'_N$
4. Compute all **pairwise distances** $d(z_i, z_j) = -\dfrac{z_i^T z_j}{\|z_i\| \|z_j\|}$
5. **Update** θ w.r.t. loss $\mathcal{L}_{\text{N-way}}(\theta) = -\sum_i \log \dfrac{\exp(-d(\tilde{z}_i, \tilde{z}'_i))}{\sum_{j \neq i} \exp(-d(\tilde{z}_i, \tilde{z}_j))}$

Additional projection head improved performance
(@ the cost of additional params (hparams))



After Pre-Training: train classifier on top of representation or fine-tune entire network.

Data imbalance may negatively affect the performance.
⇒ May be difficult to notice esp. for unlabeled datasets.

Chen, Kornblith, Norouzi, Hinton. SimCLR. ICML 2020

Performance of Contrastive Learning

ImageNet Classification Results

Semi-Supervised learning

Used x% of the overall label data → train SimCLR → do clf

Almost in few-shot learning regime

Method	Architecture	Label fraction (Top 5) 1%	10%
Supervised baseline	ResNet-50	48.4	80.4
Methods using other label-propagation:			
Pseudo-label	ResNet-50	51.6	82.4
VAT+Entropy Min.	ResNet-50	47.0	83.4
UDA (w. RandAug)	ResNet-50	-	88.5
FixMatch (w. RandAug)	ResNet-50	-	89.1
S4L (Rot+VAT+En. M.)	ResNet-50 (4×)	-	91.2
Methods using representation learning only:			
InstDisc	ResNet-50	39.2	77.4
BigBiGAN	RevNet-50 (4×)	55.2	78.8
PIRL	ResNet-50	57.2	83.8
CPC v2	ResNet-161(*)	77.9	91.2
SimCLR (ours)	ResNet-50	75.5	87.8
SimCLR (ours)	ResNet-50 (2×)	83.0	91.2
SimCLR (ours)	ResNet-50 (4×)	**85.8**	**92.6**

Table 7. ImageNet accuracy of models trained with few labels.

width of the hidden layer is 4x larger.

1% labels: ~12.8 images/class

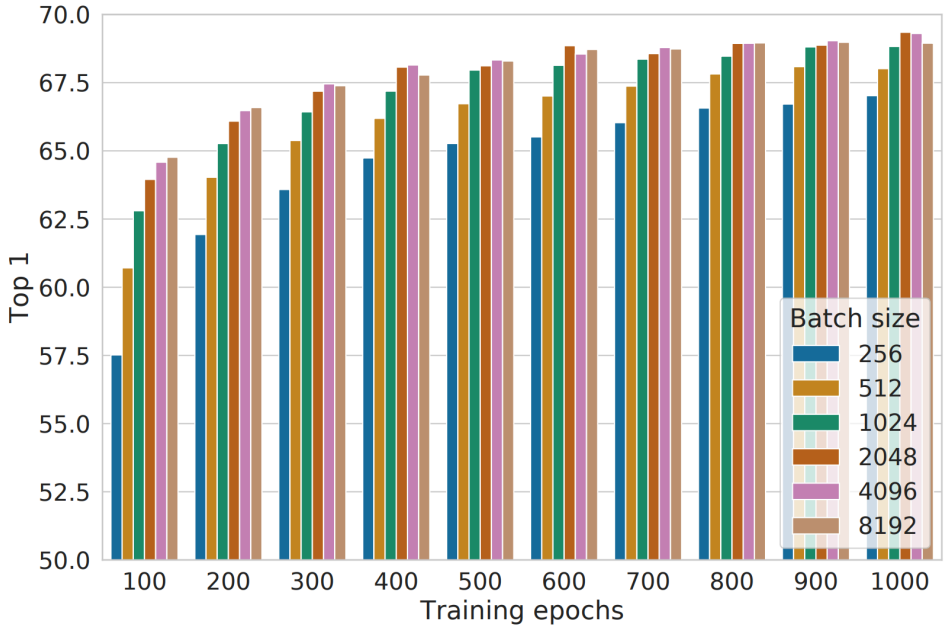
10% labels : ~128 images/class

but also uses the entire rest of the images (w/o labels)

- Substantial improvements over training from scratch
- Improvements over other methods, especially in 1% label setting

Performance of Contrastive Learning

Effect of Batch Size & Number of Training Epochs



- Important to train for longer (~600+ epochs)
- Requires large batch size

Chen, Kornblith, Norouzi, Hinton. SimCLR. ICML 2020

Why does contrastive learning need a large batch size?

Interpretation of loss: classifying augmented example from rest of dataset

$$\mathcal{L}_{\text{N-way}}(\theta) = -\sum_i \log \frac{\exp(-d(\tilde{z}_i, \tilde{z}_i'))}{\sum_{j \neq i} \exp(-d(\tilde{z}_i, \tilde{z}_j))}$$ <— summation over entire dataset

Intuition: Closest z will dominate the denominator, can be missed when subsampling

↳ applies just to $(j \neq i)$ samples → samples from different class (for instance)

Mathematically?

negatives w/ the smallest negative distances will dominate the sum

↳

$j \neq i$		
	similar	dissimilar
d :	0.01	10
	$e^{-0.01}$	e^{-10}
	(0.99)	(0.00004)

plays a much larger role

Because these examples w/ small distances are dominating, if you're subsampling (have a small batch size), you can miss the examples that actually dominate that loss func.

⇒ Need better representation of more dissimilar examples.

cf) In supervised learning, totally fine to have small batches → $-\sum_{x,y} \log P(y|x)$ Smaller batches of $x \sim y$ will still give the correct gradient. → Not the same here.

$$\min_\theta -\log \frac{e^{-d(z,z^+)}}{\sum_n e^{-d(z,z_n^-)}}$$

↳ Subsampling: sort of bringing the summation outside of the log.

$$= d(z,z^+) + \log \sum_n e^{-d(z,z_n^-)}$$

summation is inside the log.

when we subsample,

$$\approx d(z,z^+) + \sum_{\text{minibatch}} \log \sum_{n \in B} e^{-d(z,z_n^-)}$$

Jensen's Inequality

$$\log \Sigma x \geq \Sigma \log x$$

$$\therefore d(z,z^+) + \log \sum_n e^{-d(z,z_n^-)}$$

$$\geq d(z,z^+) + \sum_{\text{minibatch}} \log \sum_{n \in B} e^{-d(z,z_n^-)}$$

∴ when we minimize over the minibatch objective, we're minimizing the LB (not the UB) on our original objective.

→ <u>not good</u> → may not be minimizing on the original objective.

↳ Larger BS → closer to minimizing on the original objective.

Why does contrastive learning need a large batch size?

Interpretation of loss: classifying augmented example from rest of dataset

$$\mathscr{L}_{\text{N-way}}(\theta) = -\sum_i \log \frac{\exp(-d(\tilde{z}_i, \tilde{z}_i'))}{\sum_{j \neq i} \exp(-d(\tilde{z}_i, \tilde{z}_j))}$$ <— summation over entire dataset

Intuition: Closest z will dominate the denominator, can be missed when subsampling

Mathematically: Minimizing a lower-bound. 😱

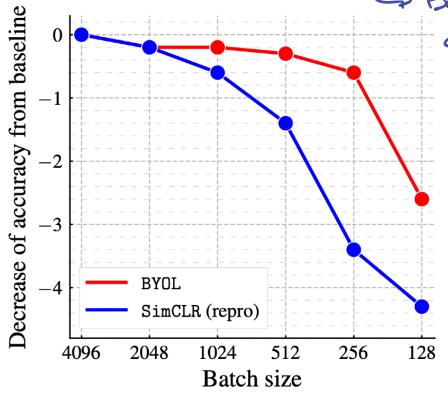
Solutions to requiring a large batch size

1. Store representations from previous batches ("momentum contrast")
   He, Fan, Wu, Xie, Girshick. MoCo. CVPR 2020
   - Good results with mini batch size of 256

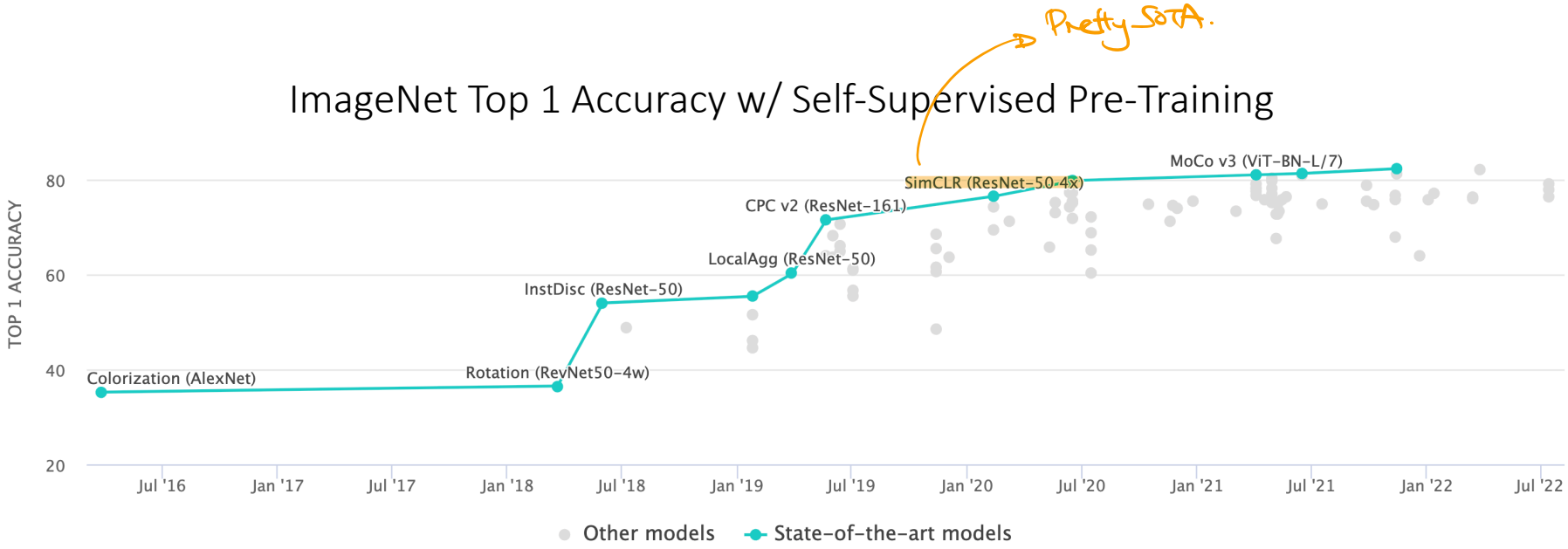
↳ Not exactly correct ∵ encoder will change over the course of training, ∴ previous representations will be outdated.
↳ but still allows (some) decoupling of BS from the estimate

2. Predict representation of same image under different augmentation ("BYOL")
   Grill*, Strub*, Altché*, Tallec* Richemond*, et al. BYOL. NeurIPS 2020
   - No negatives required!
   - More resilient to batch size

↳ Bootstrap your own latent



Performance of contrastive learning



Contrastive methods are **near state-of-the-art** in self-supervised pre-training for visual data.

Plot source: paperswithcode.com

Contrastive learning beyond augmentations

We don't have good engineered augmentations for many applications!

1. ***Learn*** the augmentations in adversarial manner (but perturbations bounded to ℓ_1 sphere)

   Tamkin, Wu, Goodman. Viewmaker Networks. ICLR 2021

   —> competitive with SimCLR on image data

   —> good results on speech & sensor data

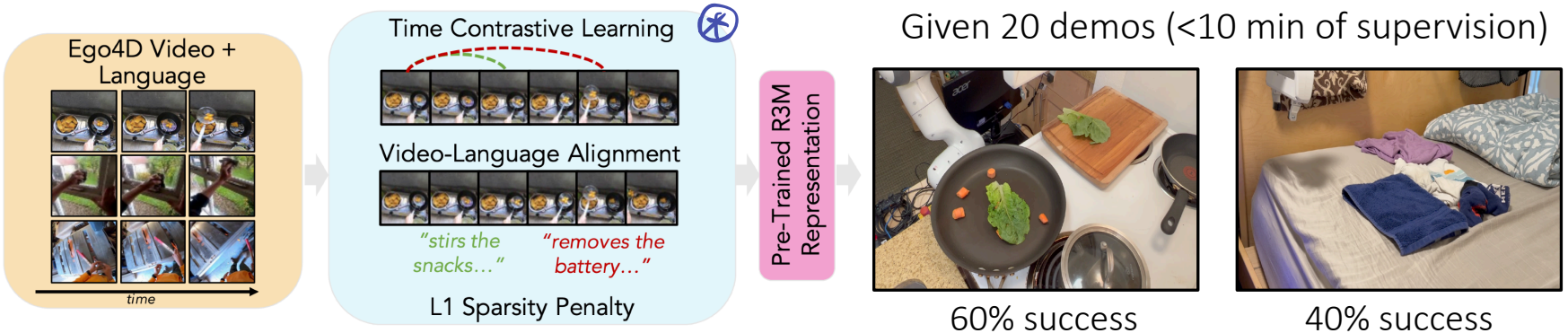
Bounded to some amount (otherwise it'll give completely arbitrary images)

Interesting

2. ***Time-contrastive learning*** on ***videos*** effective for robotics pre-training

   Nair, Rajeswaran, Kumar, Finn, Gupta. R3M. CoRL 2022.

⊗ (+)ves : Frames closer in time
(-)ves : Frames farther apart in time, or from other videos.



Contrastive learning beyond augmentations

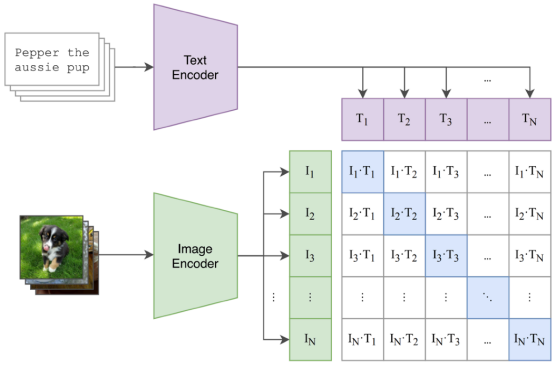
We don't have good engineered augmentations for many applications!

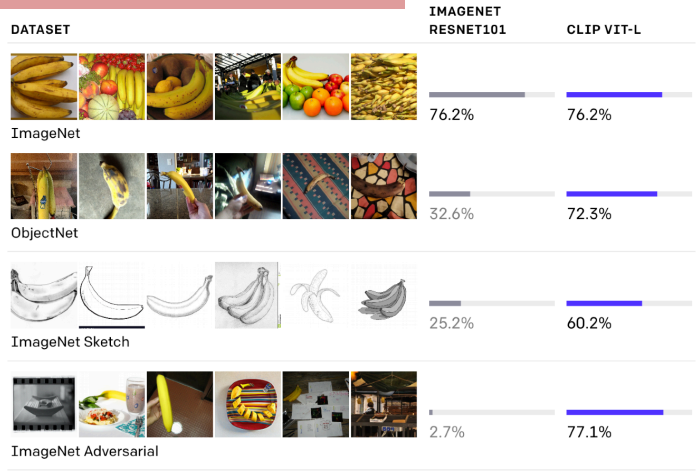
1. ***Learn*** the augmentations in adversarial manner (but perturbations bounded to ℓ_1 sphere)
   Tamkin, Wu, Goodman. Viewmaker Networks. ICLR 2021

2. ***Time-contrastive learning*** on ***videos*** effective for robotics pre-training
   Nair, Rajeswaran, Kumar, Finn, Gupta. R3M. CoRL 2022.

Apply CL btw image + txt
⇒ Saw this before somewhere

3. **Image-text** contrastive pre-training produces robust zero-shot models
   Radford*, Kim*, et al. CLIP. 2021.





Summary of Contrastive Learning

Pros:

+ General, effective framework
+ No generative modeling required
+ Can incorporate domain knowledge through augmentations

Challenges:

- *Negatives* can be hard to select
- Often requires *large batch size*
- Most successful with augmentations

This Lecture

Unsupervised representation learning for few-shot learning

Part I: Contrastive learning

Part II (next time): Reconstruction-based methods

Relation to meta-learning.

Contrastive Learning as Meta-Learning

Meta-learning algorithm

1. Given unlabeled dataset $\{x_i\}$.
2. Create image class y_i from each datapoint via data augmentation $\mathscr{D}_i := \{\tilde{x}_i, \tilde{x}_i', \dots\}$
3. Run your favorite meta-learning algorithm.

Differences:

- SimCLR samples **one task** per minibatch; meta-learning usually samples **multiple**
- SimCLR compares **all pairs** of samples; meta-learning compares query examples only to support examples & not to other query examples.

↳ SimCLR is more data-efficient w/ the batch

Otherwise the two learning methods are v. similar

Contrastive Learning as Meta-Learning

Meta-learning algorithm

1. Given unlabeled dataset $\{x_i\}$.
2. Create image class y_i from each datapoint via data augmentation $\mathcal{D}_i := \{\tilde{x}_i, \tilde{x}_i', \dots\}$
3. Run your favorite meta-learning algorithm.

comparing the repr.

Contrastive vs. meta-learning representations, transfer from ImageNet

	Flowers102	DTD	VOC2007	Aircraft	Food101	SUN397	CIFAR-10	CIFAR-100
SimCLR	92.4	72.7	66.0	83.7	86.3	57.4	94.8	79.1
ProtoNet	92.7	71.5	64.7	83.9	86.2	56.4	96.0	79.1
R2-D2	**94.5**	**73.8**	**69.9**	**86.2**	**86.9**	**59.7**	**96.7**	**82.8**

Non-parametric meta learner

optim-based meta learner

Representations transfer similarly well.

[Few-Shot Learning]: Give it a few examples, embed, ⇝ make clf. (@ test time)

[Contrastive Learning]: Learn the representation, ⇝ fine-tune the whole network (@ test time)

Ni, Shu, Souri, Goldblum, Goldstein. ICLR 2022

Lecture Outline

Unsupervised representation learning for few-shot learning

Part I: Contrastive learning

Part II (next time): Reconstruction-based methods

^ next lecture by **TA Eric Mitchell**

(NLP PhD student)

Relation to meta-learning.

Goals for the lecture:

- Understand **contrastive learning**: intuition, design choices, how to implement
- How contrastive learning relates to meta-learning

Course Reminders

Project proposal due Wednesday.

(graded lightly, for your benefit)

Homework 2 due next Monday 10/24.