DeepMind

Learning and transferring representations with few labels: BYOL & CrossTransformer

Carl Doersch

Why transfer matters

Web Data (e.g. ImageNet)


Tricycle


Hot Air Balloon


Patas Monkey


Wine Bottle


Pay Phone

. . .

General vision tasks


How to grasp the toothbrush?


Is this bridge safe?


Where does the beam go?

Is big data enough?

- “There’s billions of images on the web--let’s use them all!”
- But this doesn’t work yet
- Learning on ImageNet is good enough

	ImageNet top-1 accuracy
Resnet-200 Supervised	80.2
Resnet-33 trained on CPC features	83.4

- Self-supervised losses don’t just let us use unlabeled data, they help us **generalize better**

Henaff et al “Data-Efficient Image Recognition with Contrastive Predictive Coding” 2020

How do we build generalizable representations?

- For example: Novel scenes are composed of familiar things
  - Therefore, we can learn representations that that decompose into simpler pieces, such that representations of novel scenes/objects/tasks make it clear what is similar to familiar ones.
- In all cases, evaluation should be by **transfer** to novel tasks or datasets with little data/labels.

Veltkamp et al. 2001

Papers in this talk

- Bootstrap Your Own Latent (BYOL)
  - Better representation learning
- CrossTransformers: spatially-aware few-shot transfer
  - Better transfer

DeepMind

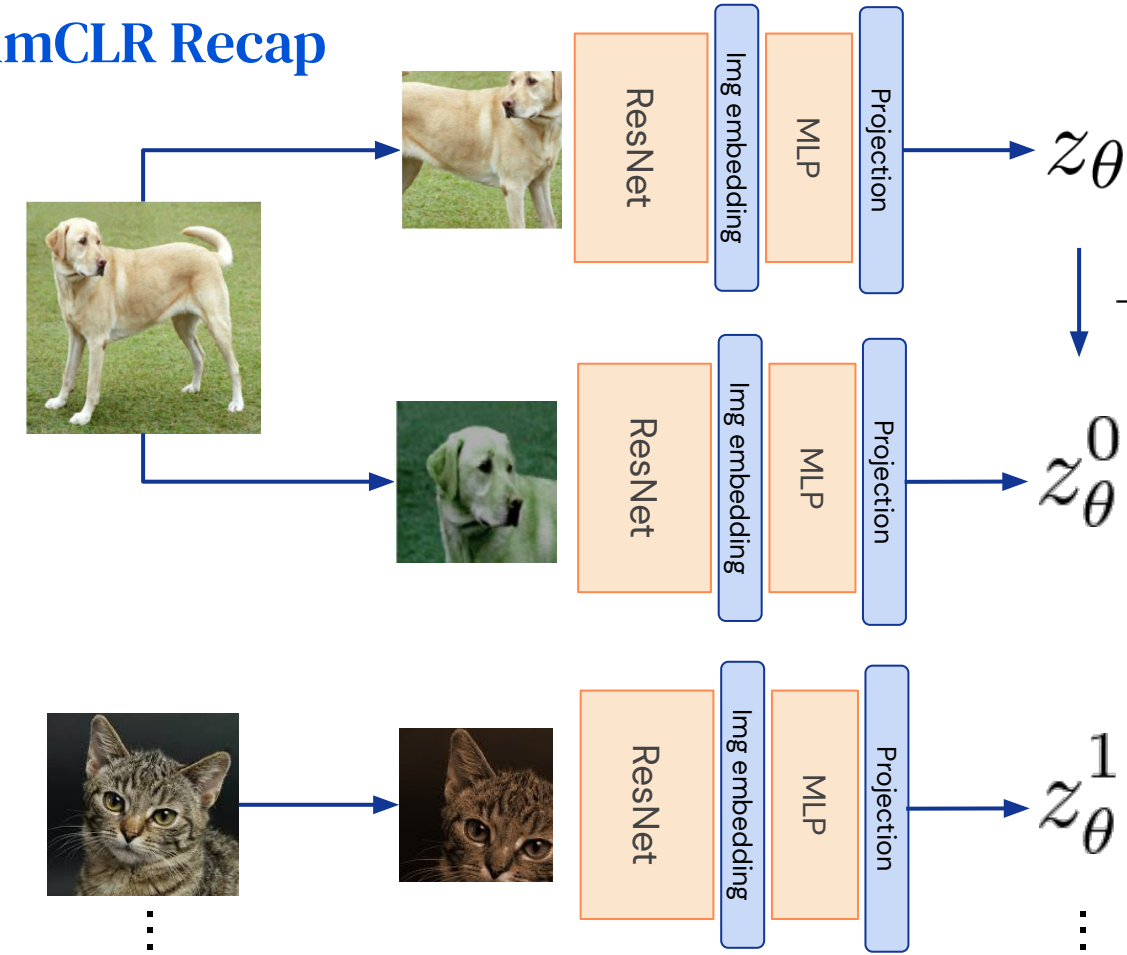
Bootstrap Your Own Latent (BYOL) : A New Approach to Self-Supervised Learning

Jean-Bastien Grill, Florian Strub, Florent Altche, Corentin Tallec, Pierre Harvey Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Remi Munos, Michal Valko.

Why does contrastive learning (e.g. SimCLR) work?

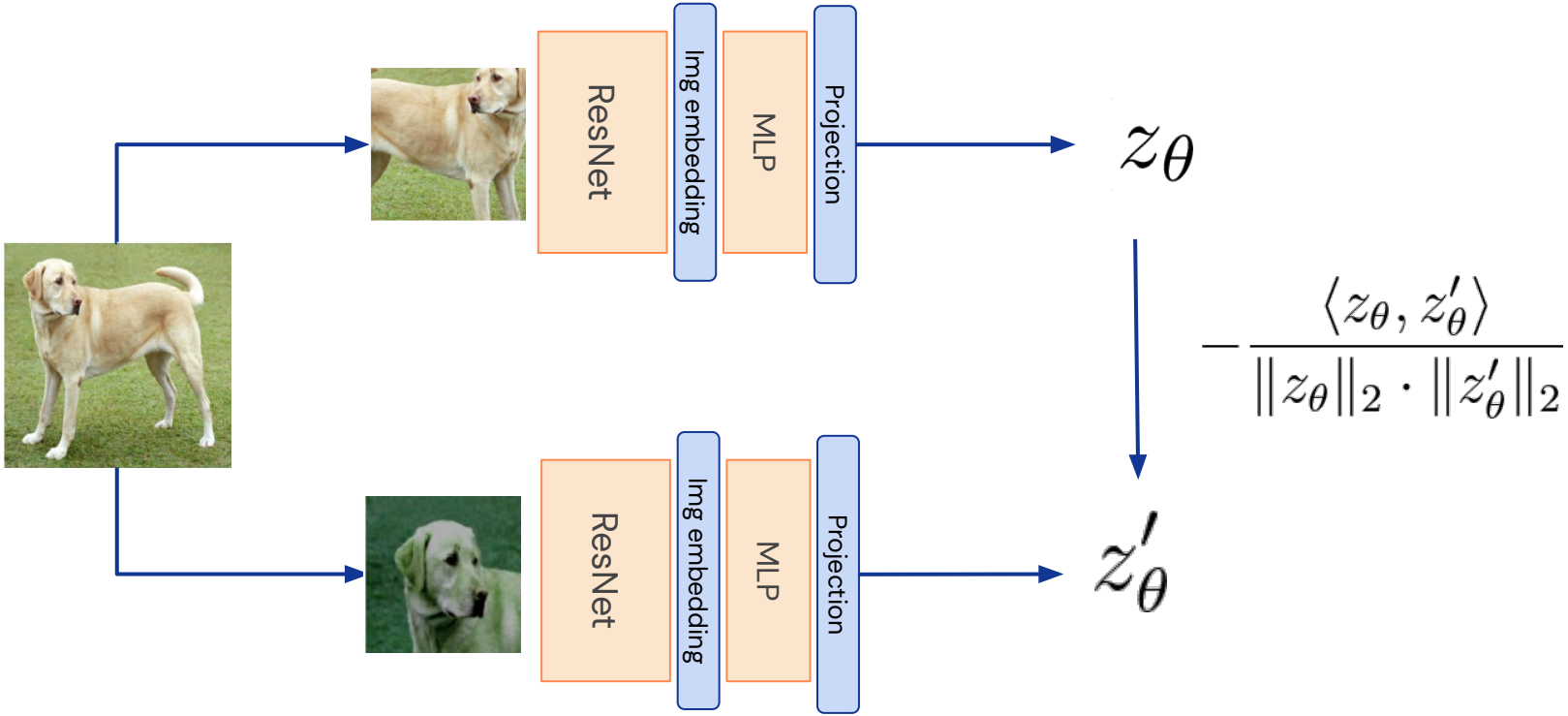
- The classic story:
  - Instance discrimination
  - Invariance to augmentation
- But are random images actually good negatives?

SimCLR Recap

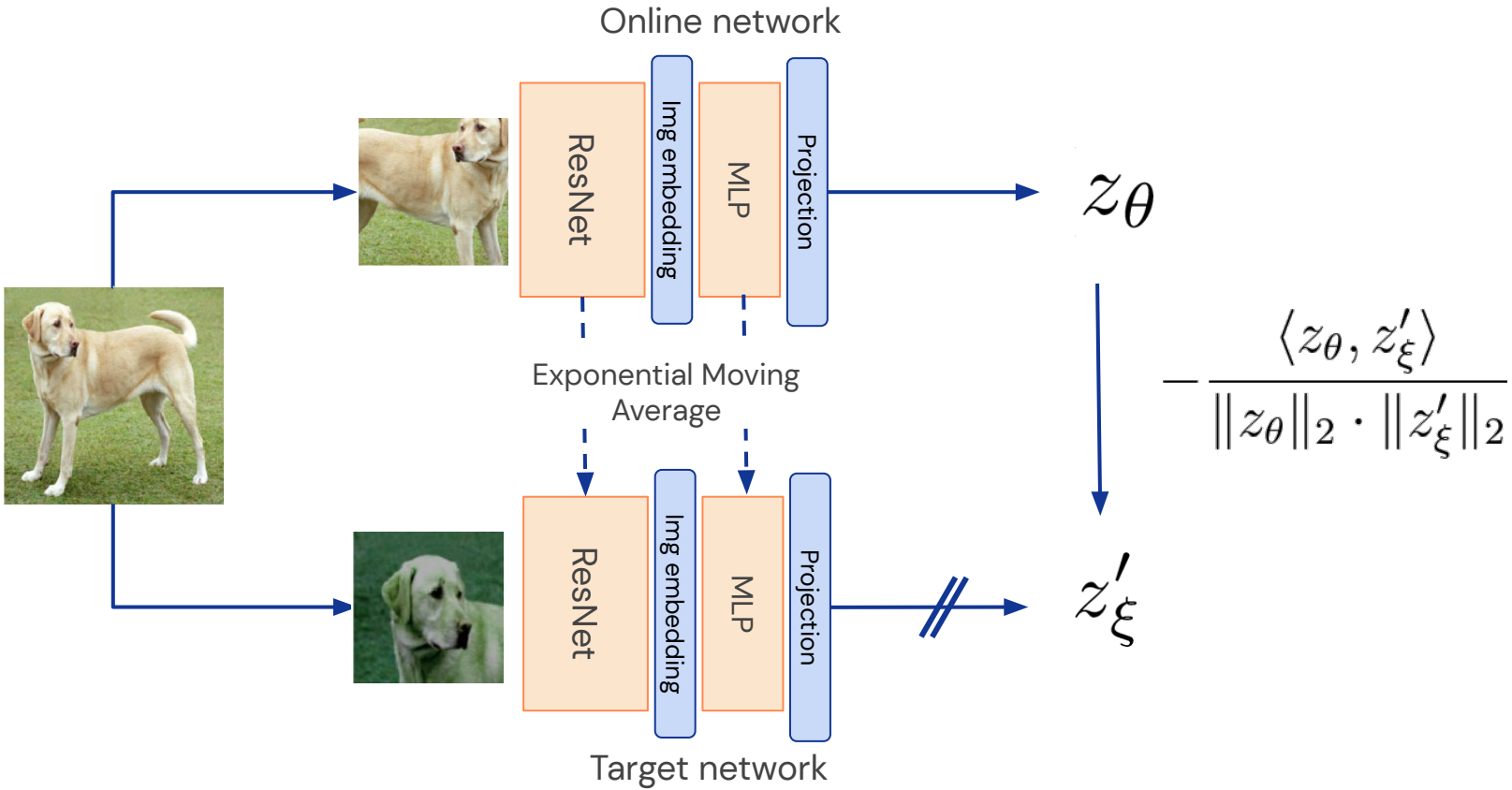


$$-\log \frac{\exp\left(\frac{\tau\langle z_\theta, z_\theta^0\rangle}{\|z_\theta\|_2 \cdot \|z_\theta^0\|_2}\right)}{\sum_i \exp\left(\frac{\tau\langle z_\theta, z_\theta^i\rangle}{\|z_\theta\|_2 \cdot \|z_\theta^i\|_2}\right)}$$

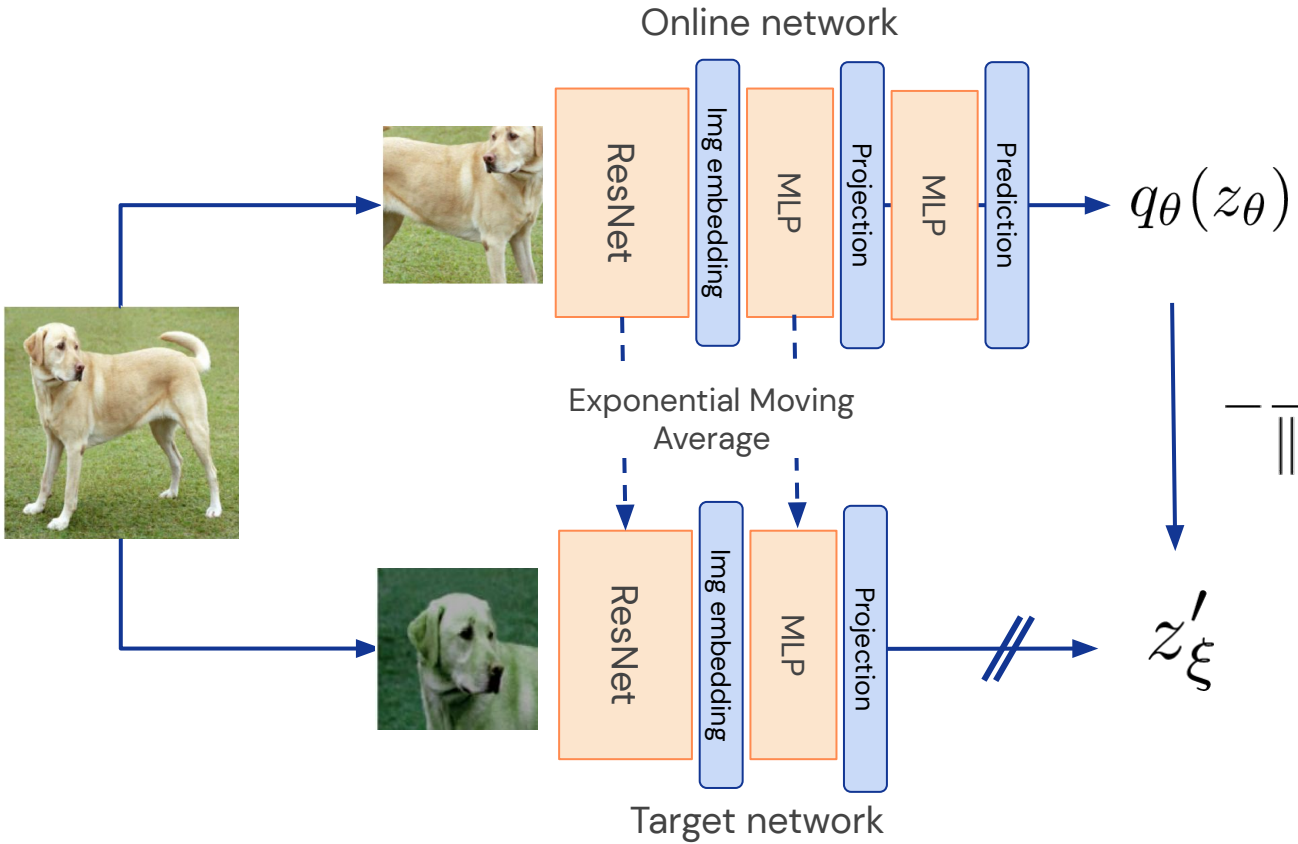
Step 1: No negatives



Step 2: Stop Gradient

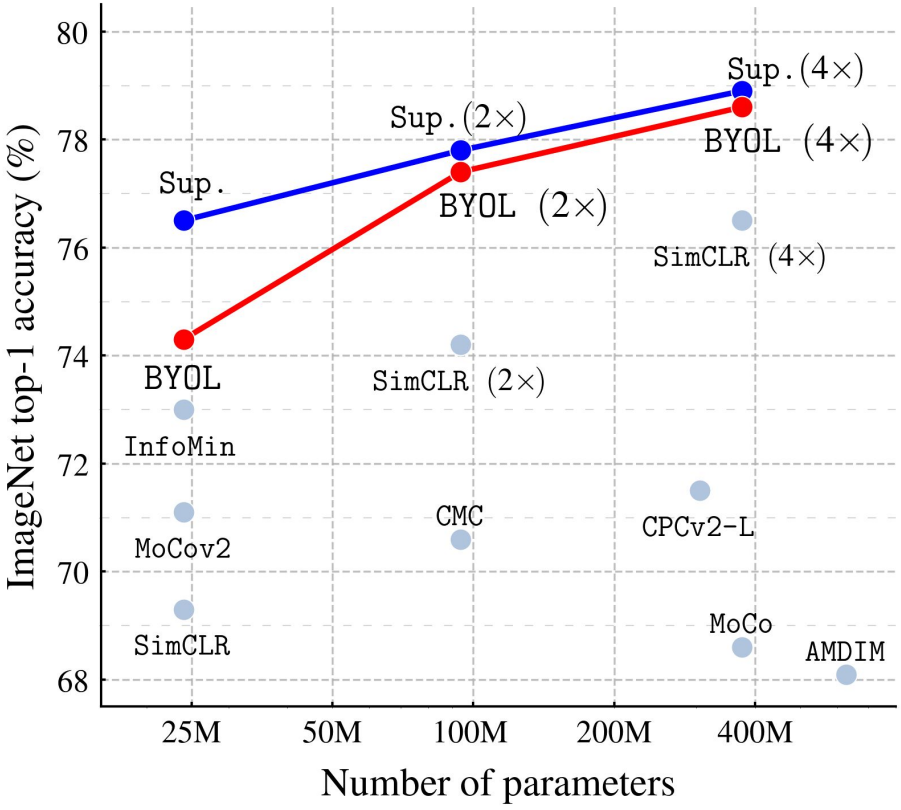


Step 3: Prediction



$$-\frac{\langle q_\theta(z_\theta), z'_\theta \rangle}{\|q_\theta(z_\theta)\|_2 \cdot \|z'_\theta\|_2}$$

Quantitative results



- SOTA on linear evaluation and semi-supervised learning on ImageNet
- Strong results on transfer: Pascal detection/segmentation, NYU Depth, Places->ImageNet

PASCAL object detection (AP) & segmentation (mIoU)

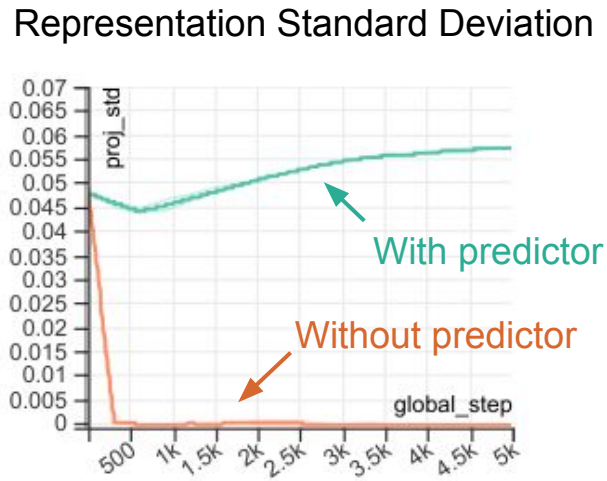
Method	AP_{50}	mIoU
Supervised-IN [9]	74.4	74.4
MoCo [9]	74.9	72.5
SimCLR (repro)	75.2	75.2
BYOL (ours)	**77.5**	**76.3**

Why does this work?

- Why doesn’t it collapse?
- What does it learn?

Why doesn't it collapse?

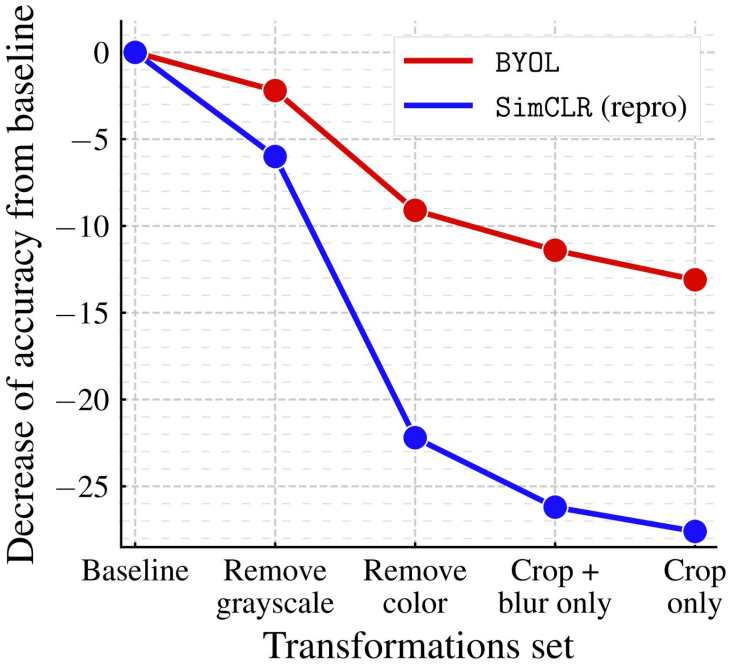
- Target network?
  - If you don't have a stop gradient, BYOL collapses
  - Skipping the moving average still works, if you increase the learning rate for the predictor
- If you don't have a predictor, BYOL collapses



Why doesn't it collapse?

- Assume the the predictor is **perfect**
- Assume the target network is **fixed**
- Then BYOL minimizes the **conditional variance** *V(target | online)*
- Increasing the **information** in the online embedding can never increase the conditional variance
- Therefore, the gradients would never reduce the information in the embedding
- Note: if the predictor is imperfect, then adding more information into z might increase **error**

What, then, does it learn?



- BYOL is less about instance discrimination and more about context prediction
- Cropping is more important for BYOL, color jittering is more important for SimCLR
- BYOL doesn't need to differentiate between all images
- BYOL's loss does not saturate when the positive pairs are similar

SimCLR

$$-\log \frac{\exp\left(\frac{\tau\langle z_\theta, z_\theta^0\rangle}{\|z_\theta\|_2 \cdot \|z_\theta^0\|_2}\right)}{\sum_i \exp\left(\frac{\tau\langle z_\theta, z_\theta^i\rangle}{\|z_\theta\|_2 \cdot \|z_\theta^i\|_2}\right)}$$

BYOL

$$-\frac{\langle q_\theta(z_\theta), z'_\theta\rangle}{\|q_\theta(z_\theta)\|_2 \cdot \|z'_\theta\|_2}$$

DeepMind

CrossTransformers: spatially-aware few-shot transfer

Carl Doersch, Ankush Gupta, Andrew Zisserman

How do we transfer representations?

- So far: fine-tuning
- Problem is formalized in **few-shot recognition**

The Challenge: represent new objects in terms of familiar ones

The task

“Query”

“Support Set”

Forster Tern



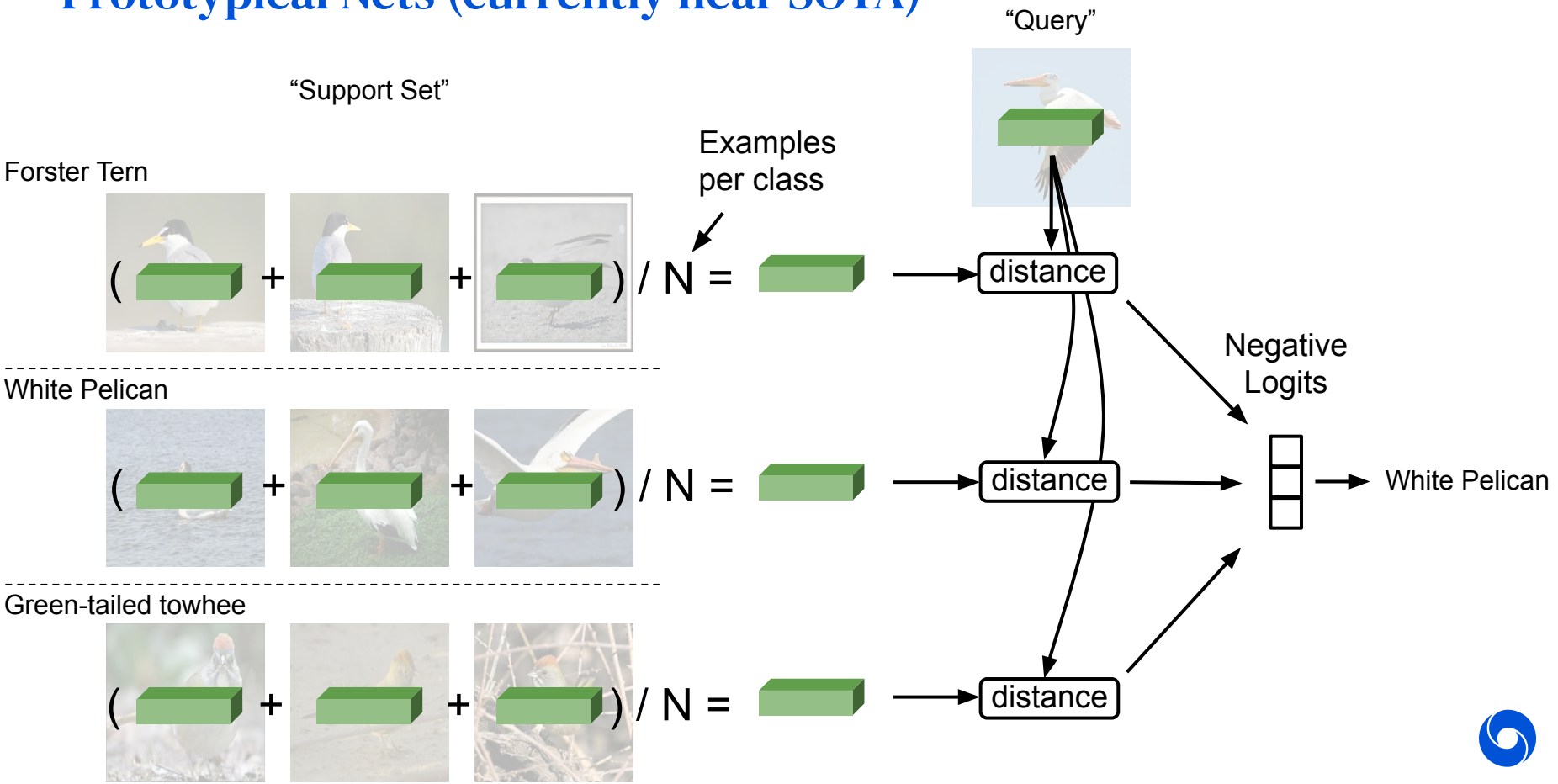
White Pelican



Green-tailed towhee



Prototypical Nets (currently near SOTA)



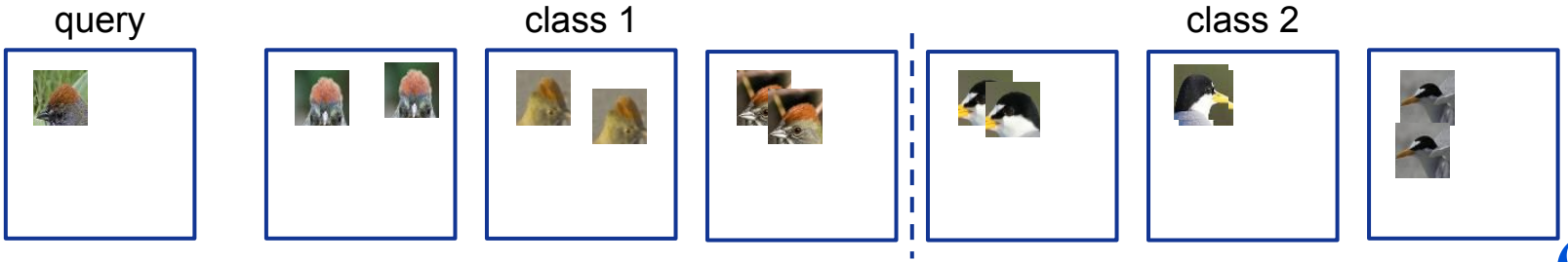
Does this work for held-out categories?

- Prototypical Nets must capture **similarity** for held out categories:
  - I.e. represent new categories in terms of familiar ones in a consistent way
- Does it work? Let's find out...
  - Split ImageNet into train and test categories
  - Train representations via Prototypical Nets on train
  - Find nearest neighbors for test-category images in both train and test sets
  - If things are working, we'll retrieve the correct test set category
  - It doesn't work...

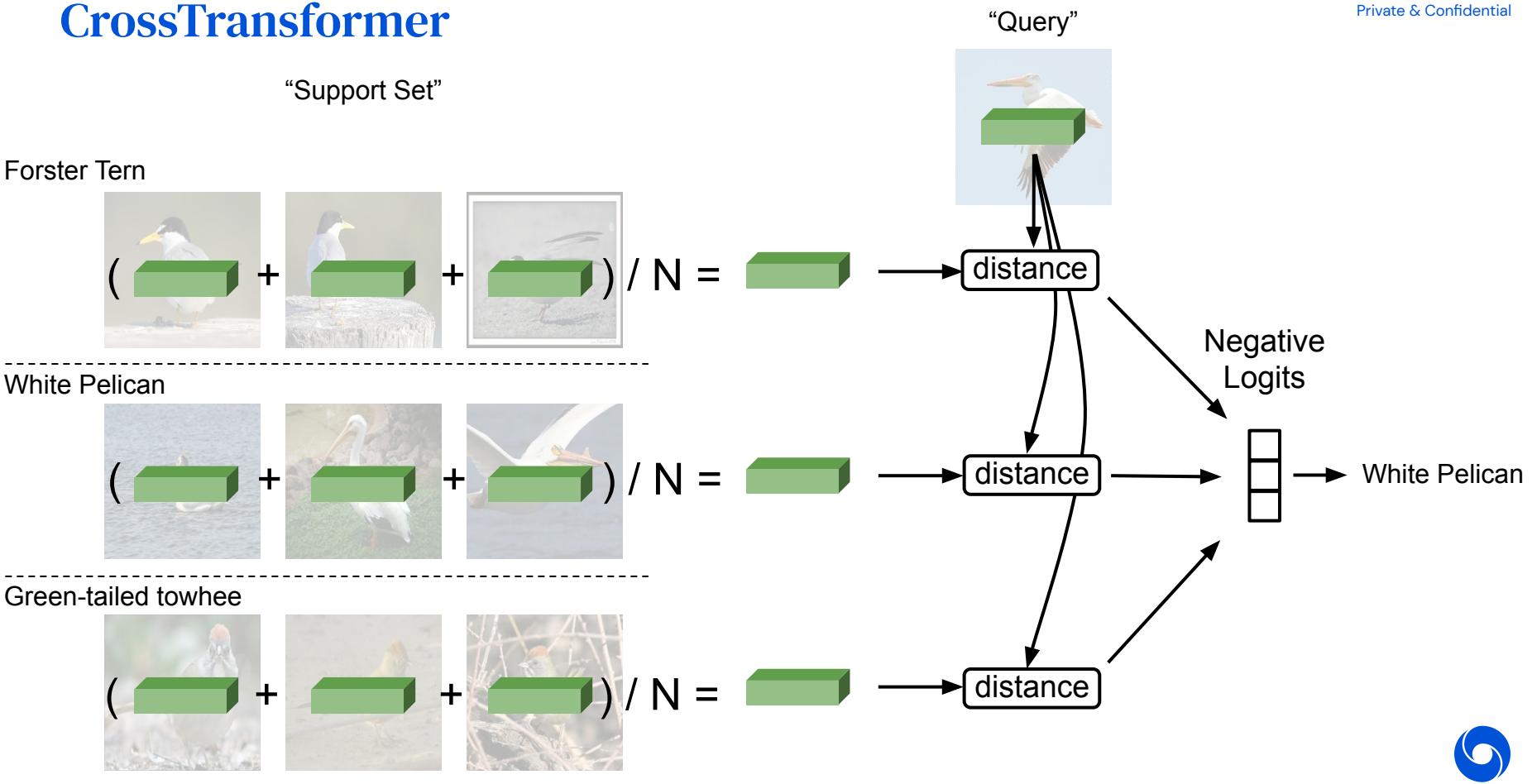
Supervision Collapse

CrossTransformers: spatially-aware comparisons

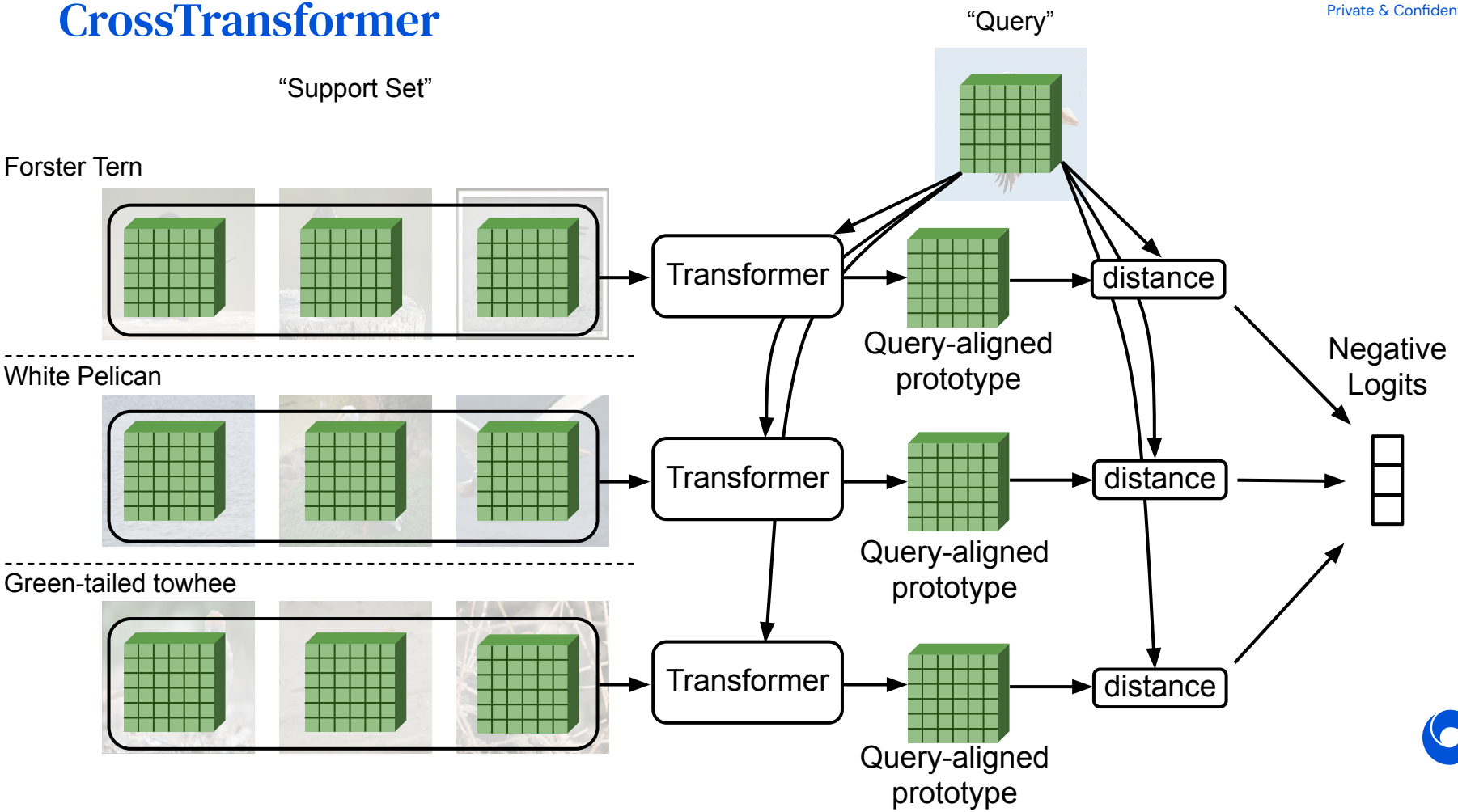
- Key idea: decompose objects into simpler, local parts that can be put into correspondence
- Then compare corresponding parts: hopefully local features are familiar



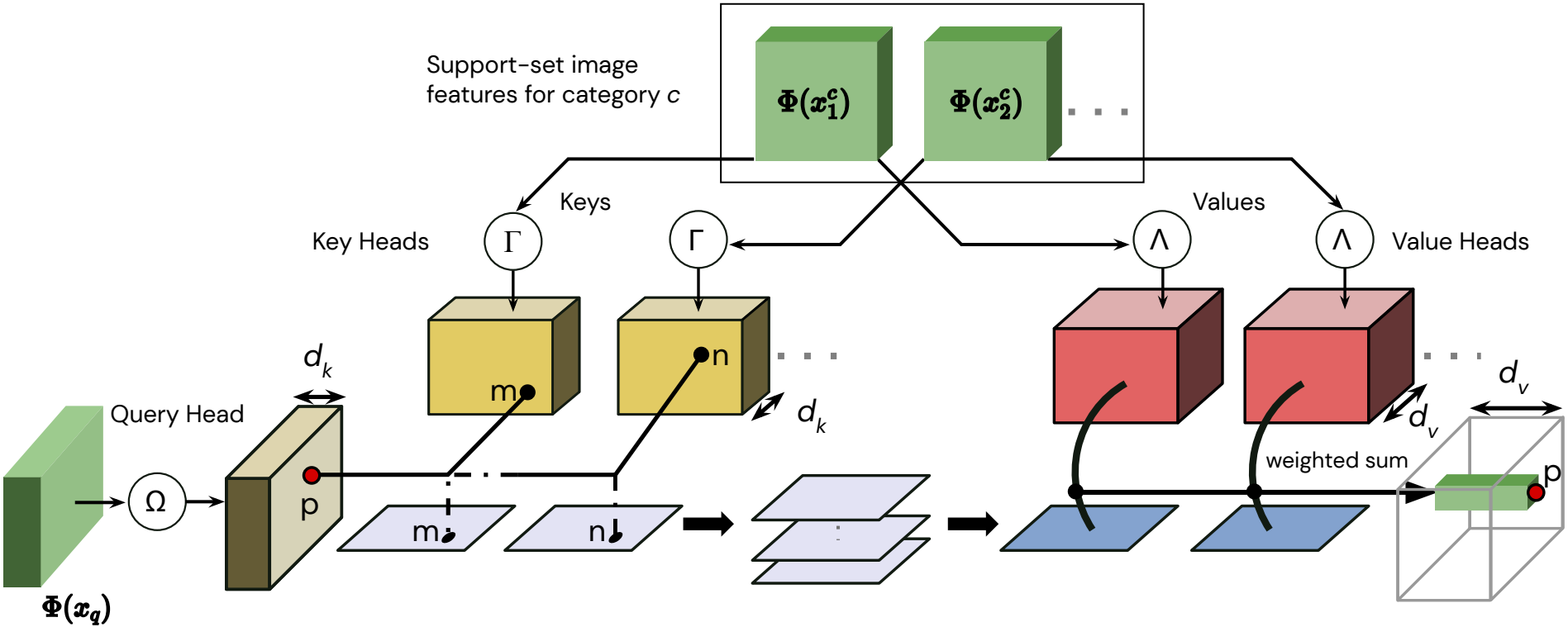
CrossTransformer



CrossTransformer



CrossTransformer: Aggregating query-aligned prototypes



Query image features

Queries

Dot-product query features at location p against all support set spatial features in class

Softmax across all spatial features in class

Softmax normalized attention weights

Query-aligned Prototype

But what if features have already collapsed to categories?

- Need to encourage features to distinguish between **instances** rather than just categories
- Instance recognition is a classic self-supervised task
  - E.g. SimCLR
- We can train for this without changing the network!
- Prototypical Nets, CrossTransformers, etc. train on “episodes” (consisting of a “support set” and “queries”)
- Train for instance recognition just by adding some “episodes” that require it: **SimCLR Episodes**

Constructing SimCLR Episodes

Forster Tern



White Pelican



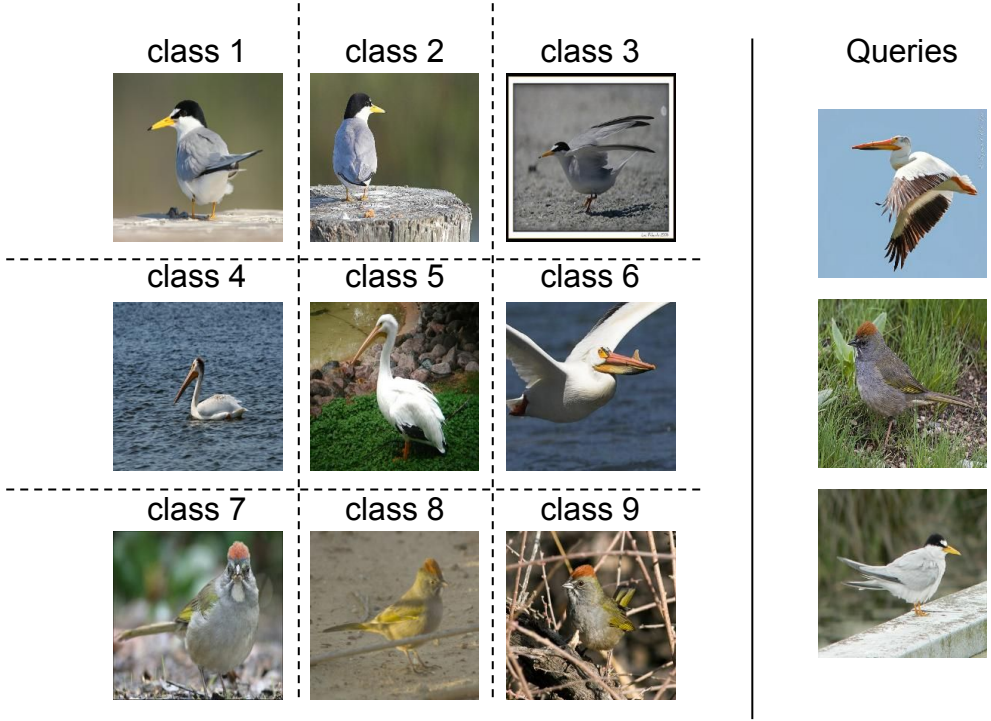
Green-tailed Towhee



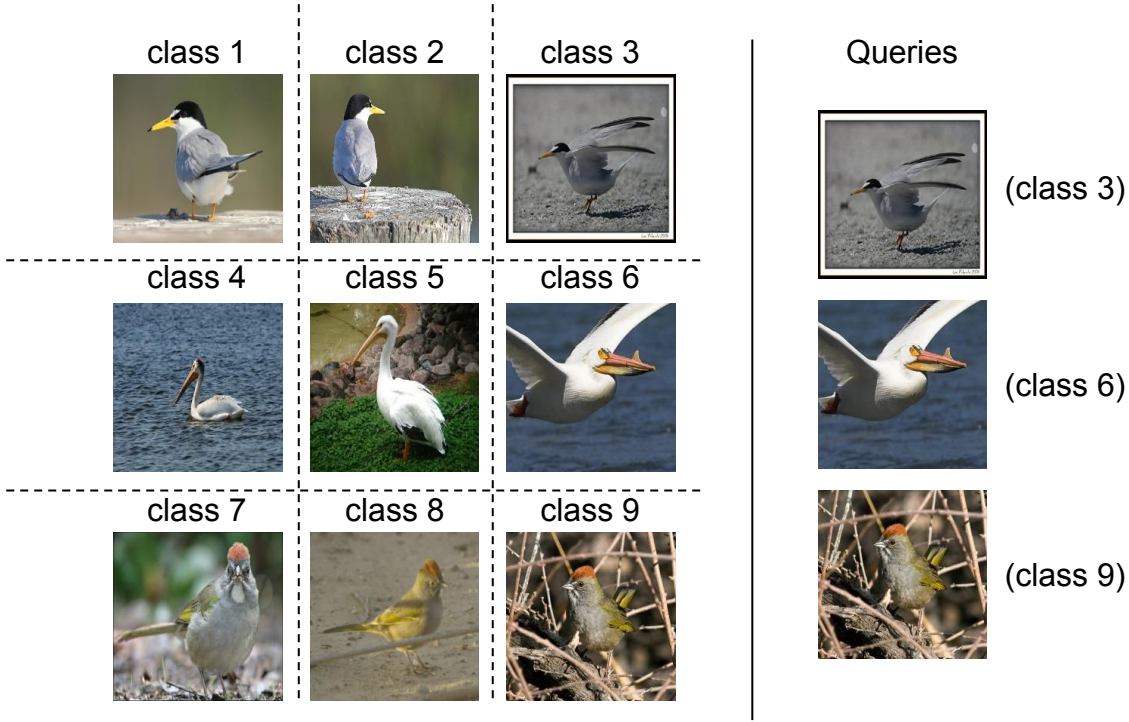
Queries



Constructing SimCLR Episodes



Constructing SimCLR Episodes



Meta-Dataset (Triantafillou et al. 2020)

- Training: a subset of ImageNet categories
- Testing: Support Sets contain 50-500 images, in 5-50 categories
- Taken from held-out fine-grained recognition datasets:


ImageNet devices
(130 classes)

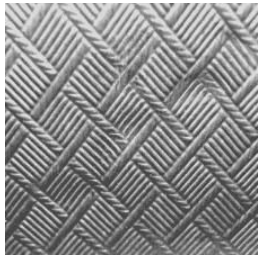

OmniGlot
(1623 classes)


Caltech Birds
(200 classes)


MSCOCO Objects
(80 classes)

Aircraft
(102 classes)


Traffic Signs
(43 classes)


DTD Textures
(47 classes)


QuickDraw
(345 classes)


Fungi
(1500 classes)


VGG Flowers
(102 classes)

Results: Qualitative

Query

Support Set Correspondences

Results: Qualitative

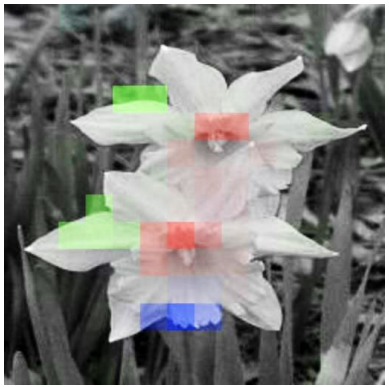
Query

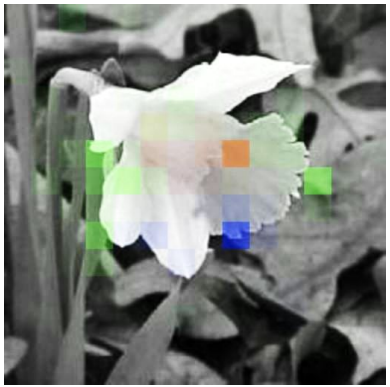
Support Set Correspondences

Results: Qualitative

Query

Support Set Correspondences

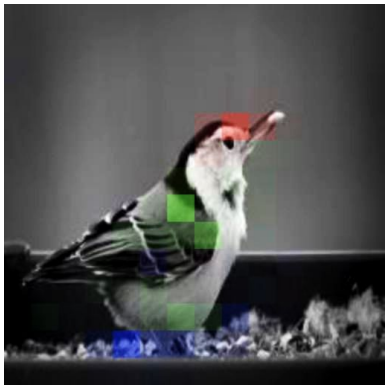




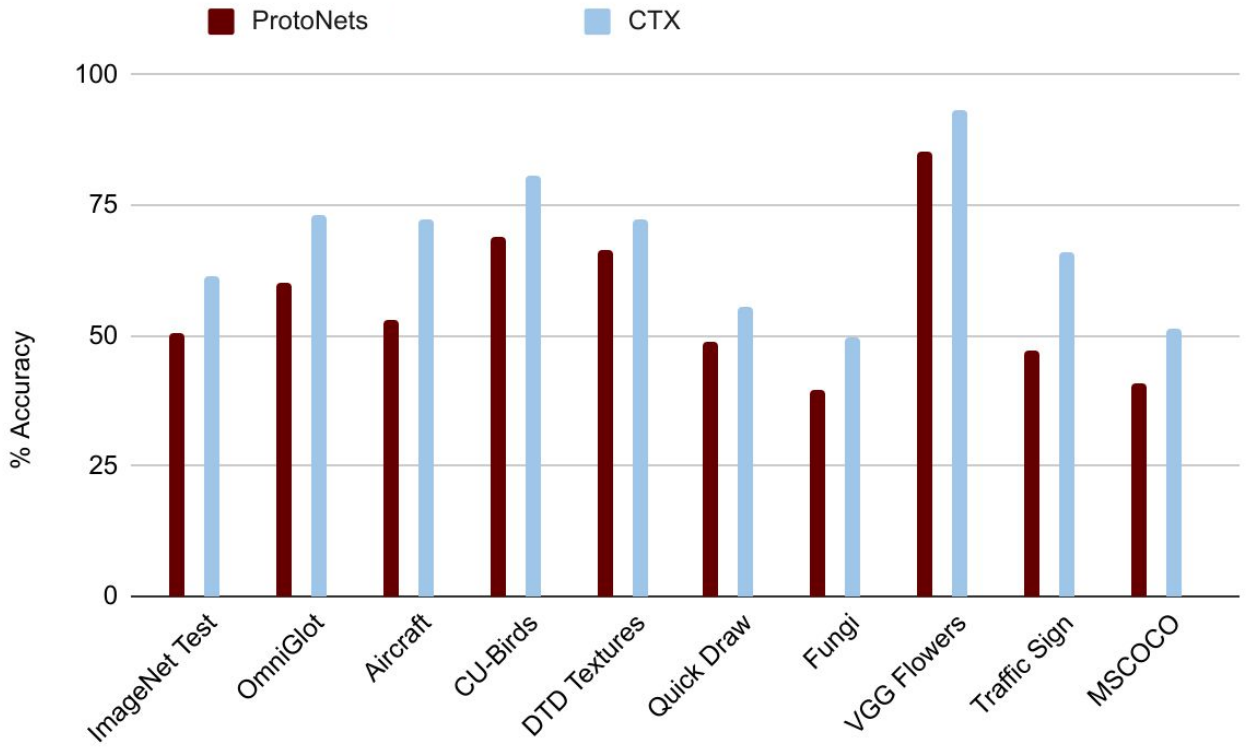
Results: Qualitative

Query

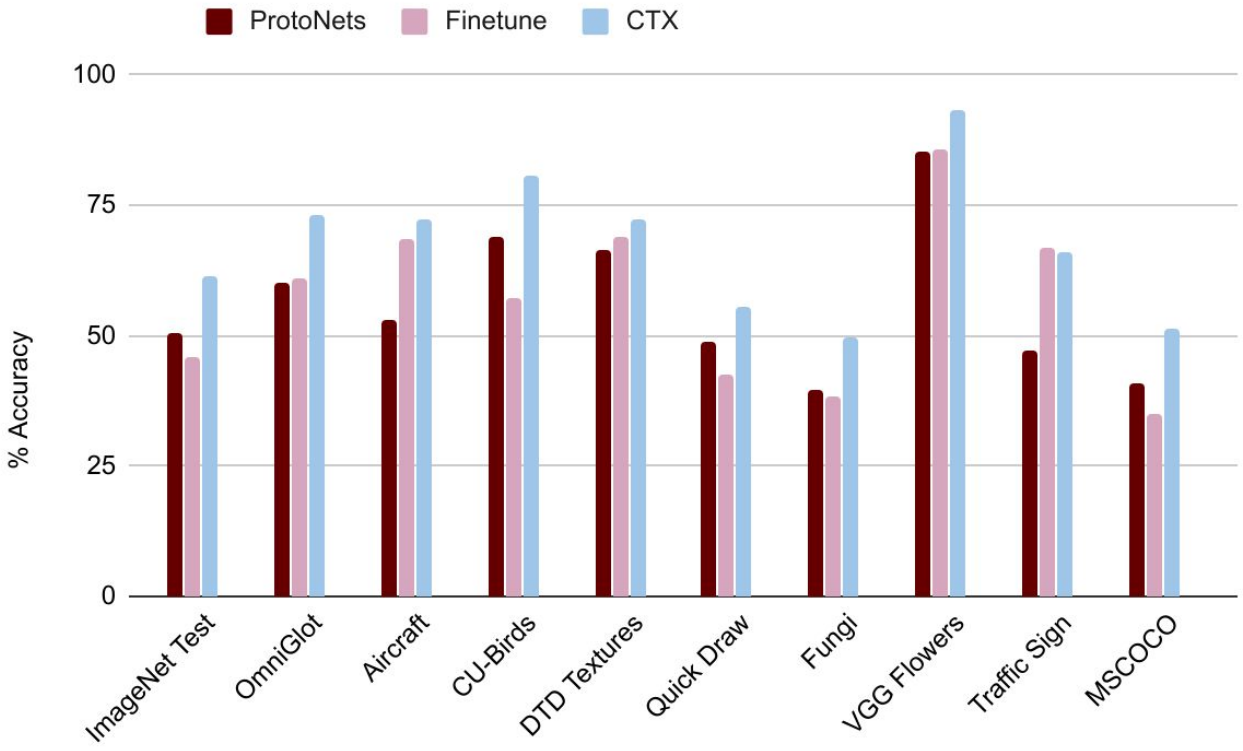
Support Set Correspondences



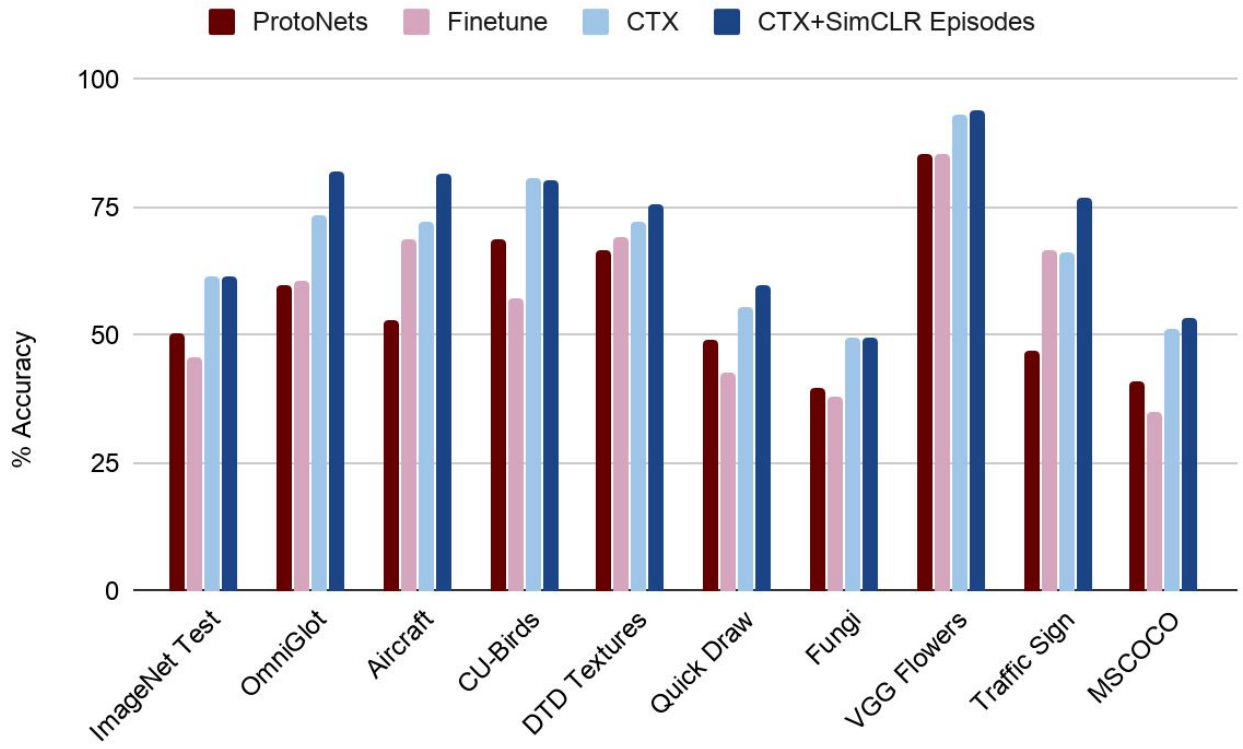
Quantitative Results: Comparison to Baselines



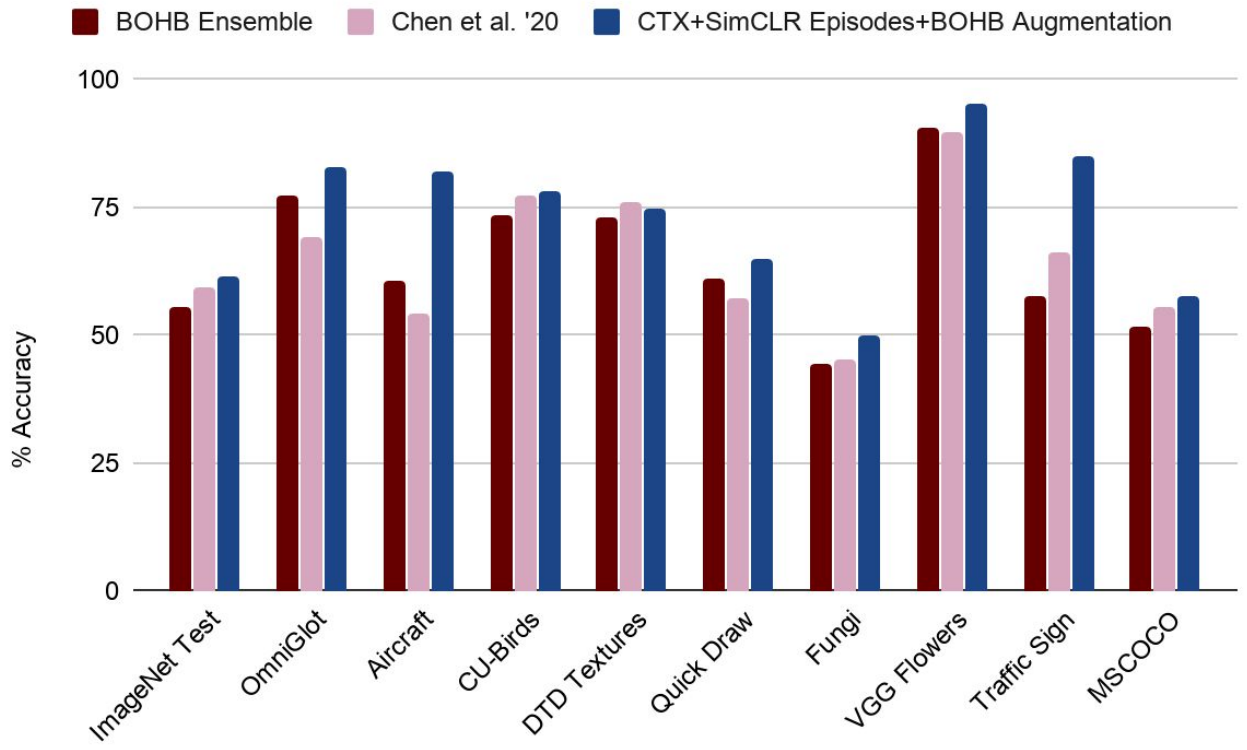
Quantitative Results: Comparison to Baselines



Quantitative Results: Comparison to Baselines



Quantitative Results: Comparison to SOTA (with augmentation)



The bigger picture

- CrossTransformers learn and exploit correspondence without explicit supervision for it; but maybe there should be losses
- SimCLR episodes use self-supervised losses to guide the features and prevent collapse

What is next?

- Transfer is especially under-studied
  - Need representations which distinguish the properties that transfer from the ones that don't
- Vision tends to pick an existing solution and hack on it
  - E.g. Linear evaluation/finetuning, contrastive methods
  - There's many physical truths that SSL could exploit, but isn't
  - Downstream task diversity is important