Self-supervised learning

682: Neural Networks: A Modern Introduction

Subhransu Maji

April 16, 2026

College of INFORMATION AND COMPUTER SCIENCES

UMASS AMHERST

Administrative

Midterm 2 on Tuesday, April 28 in class
Syllabus: Lecture 9 onwards (Image classification with CNNs)

Last Class

- Recap
  - Supervised vs Unsupervised Learning
  - Why not always label data?
- Semi-supervised Learning
  - Concepts
  - Example: pseudo-labels / self-training
- Self-supervised Learning
  - Concepts
  - Pretext tasks
  - Contrastive Learning
  - Beyond images

Recap: Supervised vs Unsupervised Learning

Supervised Learning

Data: (X, y)

X = input/feature/image/…

y = label/target

→ Cat

→ Dog

Unsupervised Learning

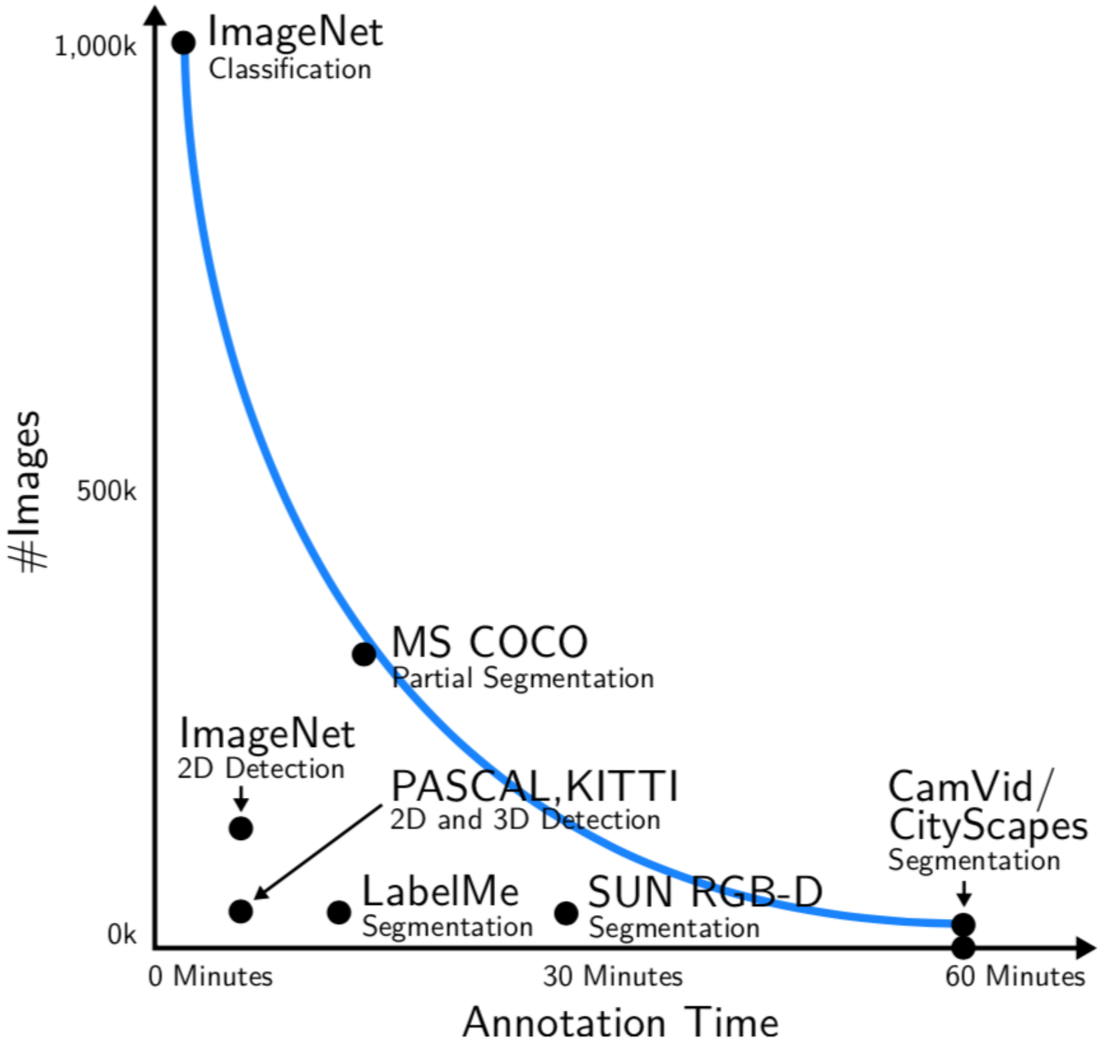
Data: X

Just X, no labels

Learn about the *structure* of the data, i.e. P(X)

…..

Recap: Annotate Everything — Expensive, doesn't Scale!



Slides from Andreas Geiger, MPI Tubingen

Recap: Motivation - Humans learn with little supervision

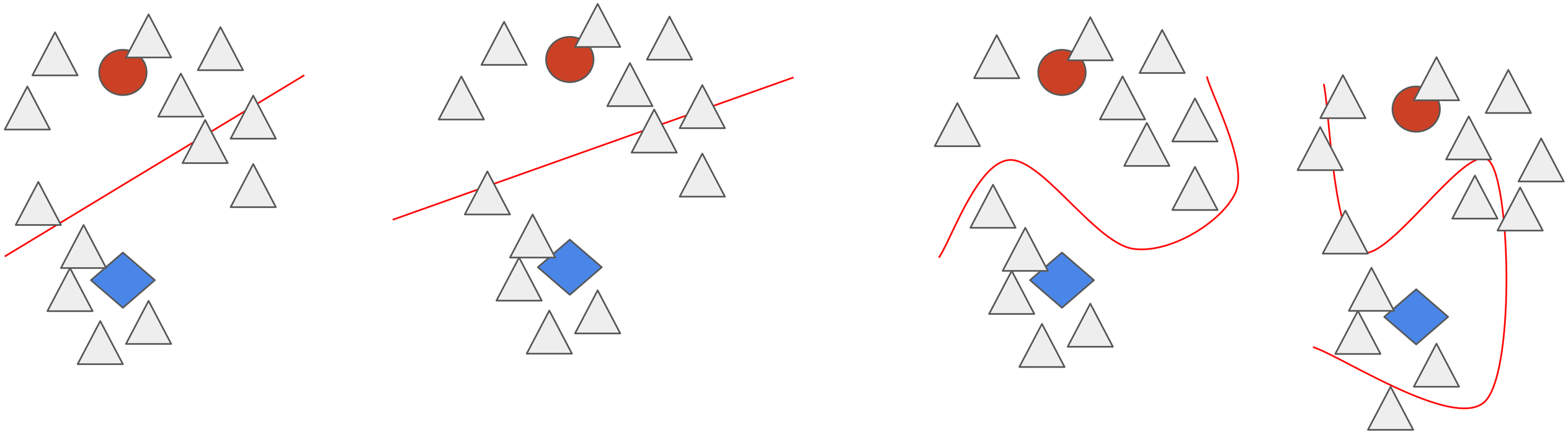
Provided with very few “labeled” examples (someone pointing something out to us explicitly), we can generalize quite well.

Slides from Andreas Geiger, MPI Tubingen

Recap: Semi-supervised Learning

- Given a small amount of ***labeled*** data $\mathcal{X}_L$
- Given (usually) large amount of ***unlabeled*** data $\mathcal{X}_U$
- Can $\mathcal{X}_U$ help us in getting a better model?

Now we see some unlabeled data points ….



Recap: Self-training

- Assume: one's own high confidence predictions are correct!

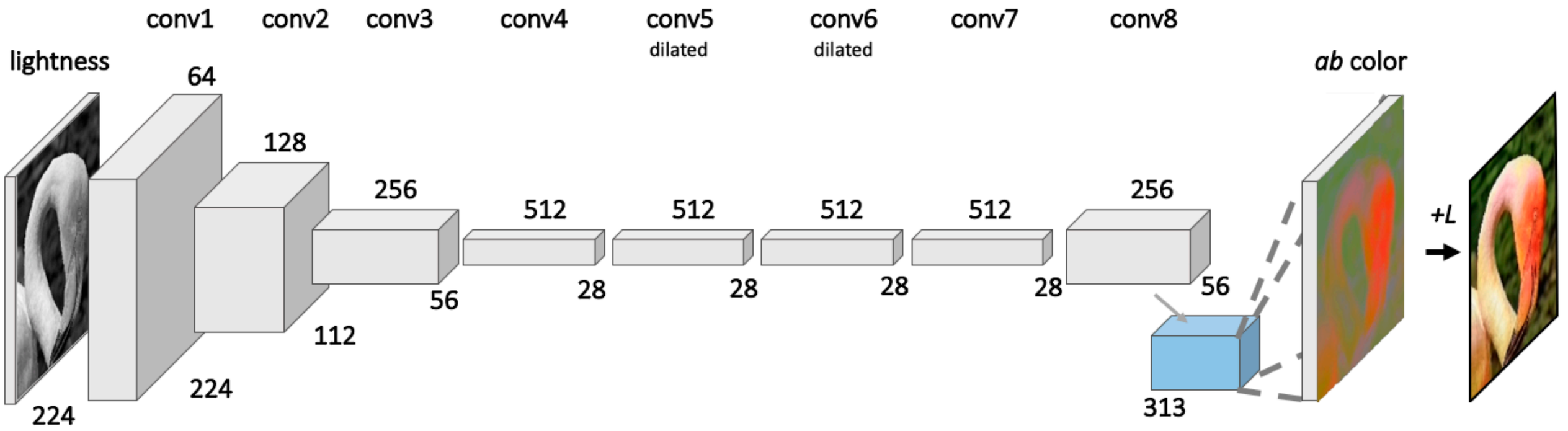
- Train model f on $\mathcal{X}_L := \{x_L, y_L\}$
- Use f to predict "pseudo-labels" on $\mathcal{X}_U := \{x_u\}$
- Add $\{x_u, f(x_u)\}$ to labeled data
- Repeat

Based off Joelle Pineau's **COMP-551**

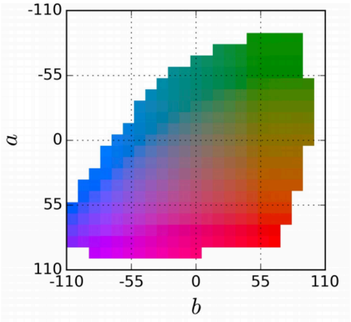
Recap: Self-supervised learning: Outline

- Data prediction
  - Colorization
- Transformation prediction
  - Context prediction, jigsaw puzzle solving, rotation prediction
- “Siamese” methods
  - Contrastive methods
  - Non-contrastive methods
- Self-supervision beyond still images
  - 3D, audio, video, language

Colorization: Architecture



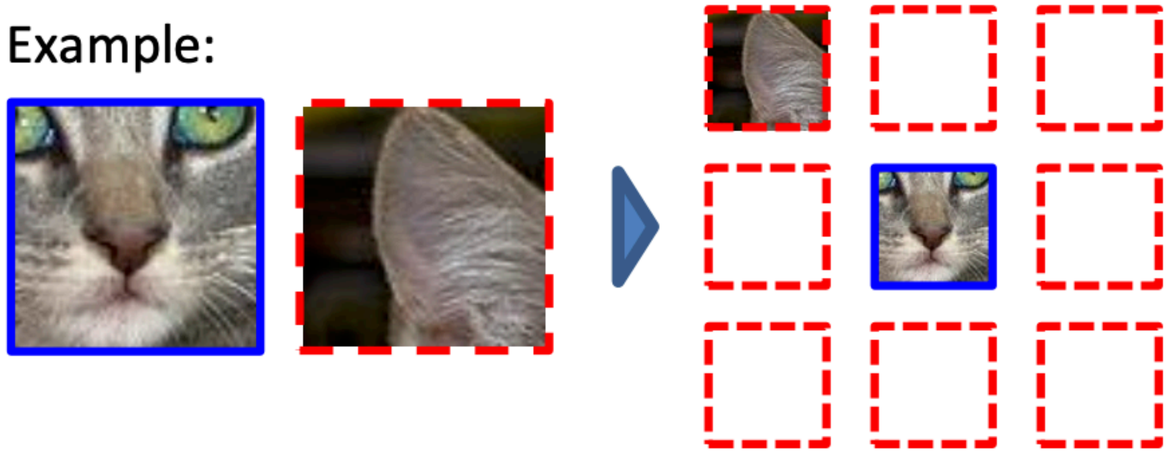
At each spatial location, predict probability distribution over 313 quantized (a,b) values



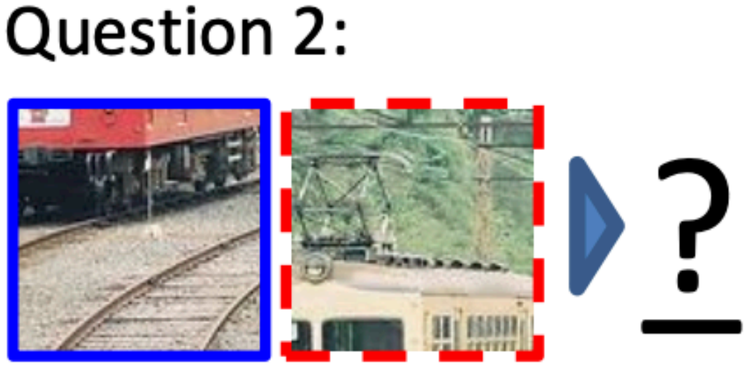
R. Zhang, P. Isola, and A. Efros, **Colorful Image Colorization**, ECCV 2016

Context prediction

- *Pretext task*: randomly sample a patch and one of 8 neighbors
- Guess the spatial relationship between the patches



A: Bottom right

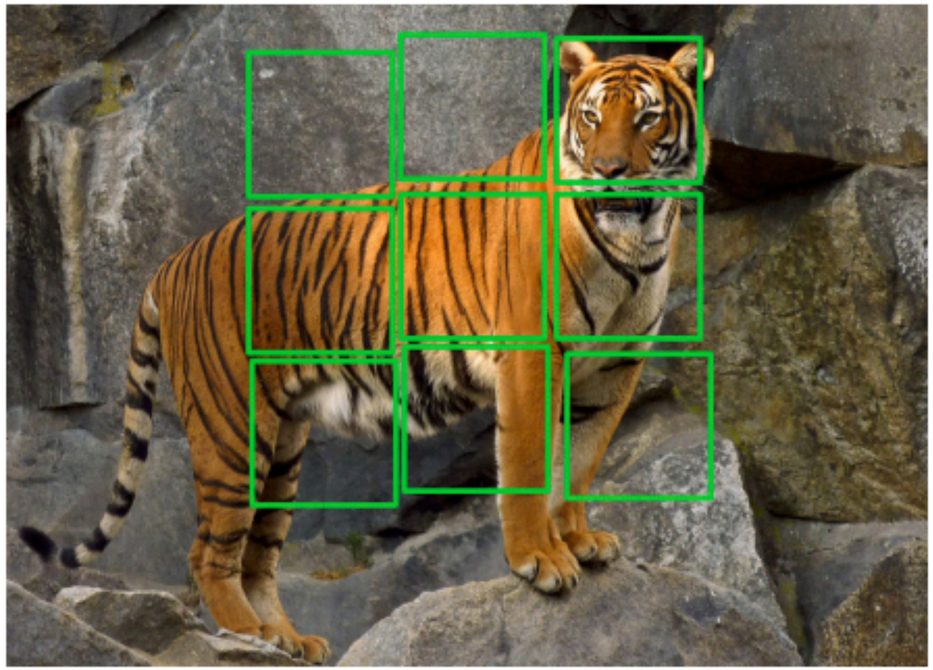


A: Top center

C. Doersch, A. Gupta, A. Efros. Unsupervised Visual Representation Learning by Context Prediction. ICCV 2015

Jigsaw puzzle solving

Crop out tiles



Shuffle

Pretext task: reassemble

Claim: jigsaw solving is easier than context prediction, trains faster, transfers better

M. Noroozi and P. Favaro. **Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles**. ECCV 2016

Rotation prediction

- Pretext task: recognize image rotation (0, 90, 180, 270 degrees)

S. Gidaris, P. Singh, and N. Komodakis. **Unsupervised representation learning by predicting image rotations**. ICLR 2018

PASCAL VOC Transfer Results

Method	Classification	Detection (mAP)	Segmentation (mIoU)
Supervised (ImageNet)	**79.9**	**56.8**	**48.0**
Colorization	65.6	46.9	35.6
Context	65.3	51.1	
Jigsaw	67.6	53.2	37.6
Rotation	73.0	54.4	39.1

Contrastive methods

- Encourage representations of transformed versions of the same image to be the same and different images to be different

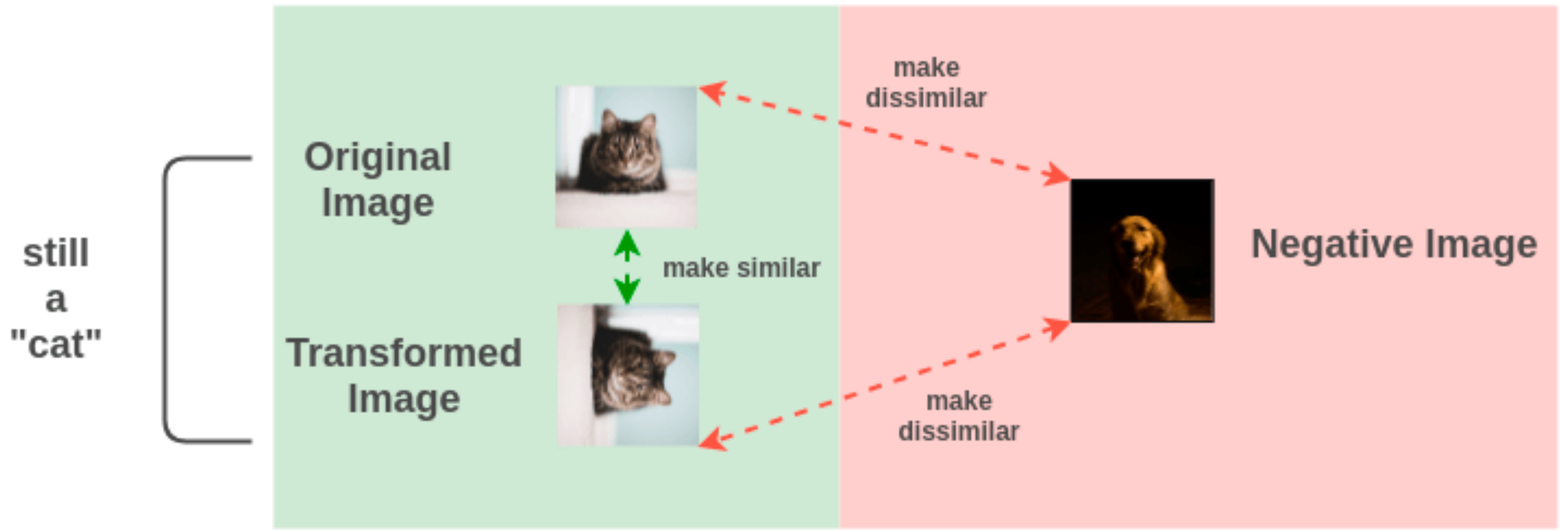


Figure source

Contrastive loss formulation

- Given:
  - *Query* point x
  - *Positive* sample x^+: version of x subjected to a random transformation or augmentation (cropping, rotation, color change, etc.)
  - *Negative* samples x^-

x x^+ x^-

Contrastive loss formulation

- Given: query x, positive sample x^+, negative samples x^-
- Measure similarity by dot product of L2-normalized feature representations:

$$\text{sim}(x, y) = \frac{f(x)}{\|f(x)\|_2} \cdot \frac{f(y)}{\|f(y)\|_2}$$

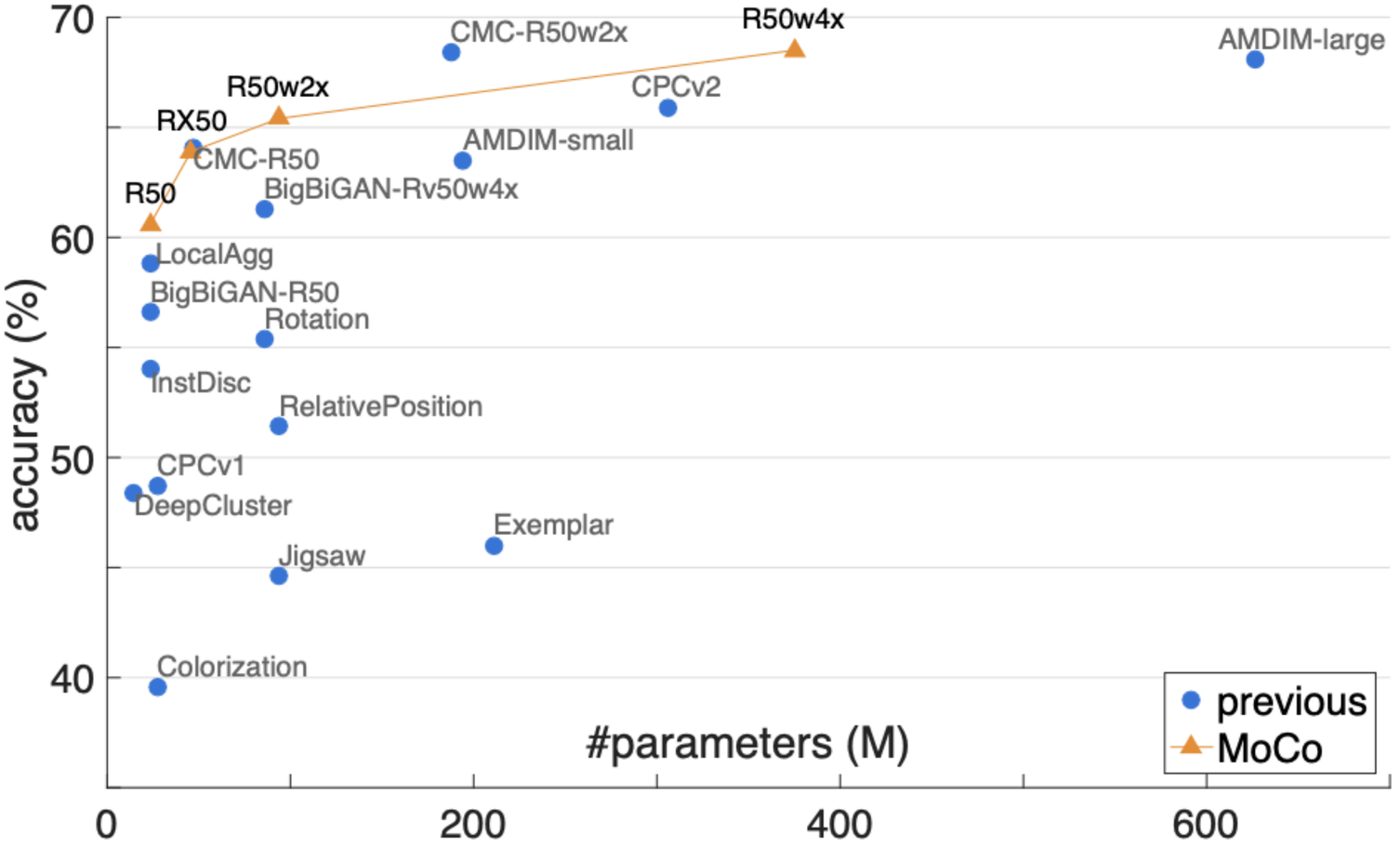
- **Contrastive loss**: make x similar to x^+, dissimilar from x^-:

$$l(x, x^+) = -\log \frac{\exp(\text{sim}(x, x^+)/\tau)}{\exp(\text{sim}(x, x^+)/\tau) + \sum_{j=1}^{N} \exp(\text{sim}(x, x_j^-)/\tau)}$$

- Intuitively, this is the loss of a softmax classifier that tries to classify x as x^+

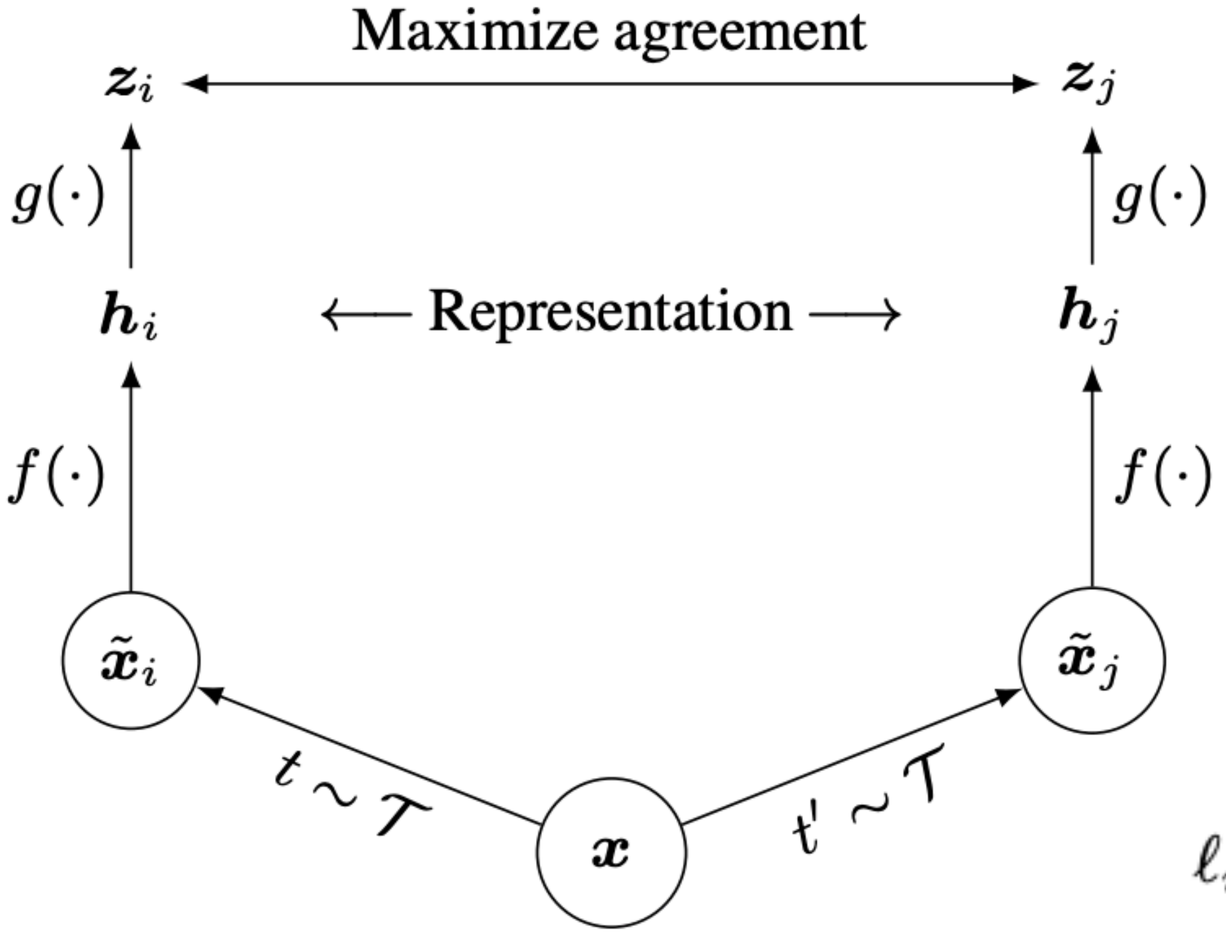
MoCo results

Comparison on linear ImageNet classification
(supervised accuracy above 75%)



K. He et al. **Momentum Contrast for Unsupervised Visual Representation Learning**. CVPR 2020

SimCLR

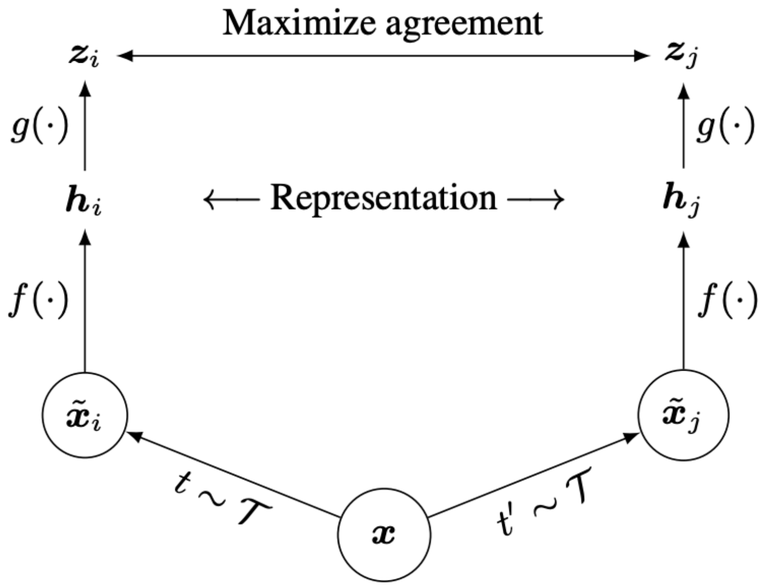


- Instead of memory bank or queue, use large mini-batch size (on cloud TPU)
- Introduce nonlinear *projection* ($_g$) between representation ($_h$) and feature used for computing contrastive loss ($_z$)

$$\ell_{i,j} = -\log \frac{\exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}$$

T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. **A Simple Framework for Contrastive Learning of Visual Representations**. ICML 2020

SimCLR



$$\ell_{i,j} = -\log \frac{\exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}$$

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 $k \in \{1, \ldots, N\}$ **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 $i \in \{1, \ldots, 2N\}$ and $j \in \{1, \ldots, 2N\}$ **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 $g(\cdot)$

T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. **A Simple Framework for Contrastive Learning of Visual Representations**. ICML 2020

SimCLR

- Performed extensive ablation study of data augmentations
- Found that composing multiple augmentations gives the best results

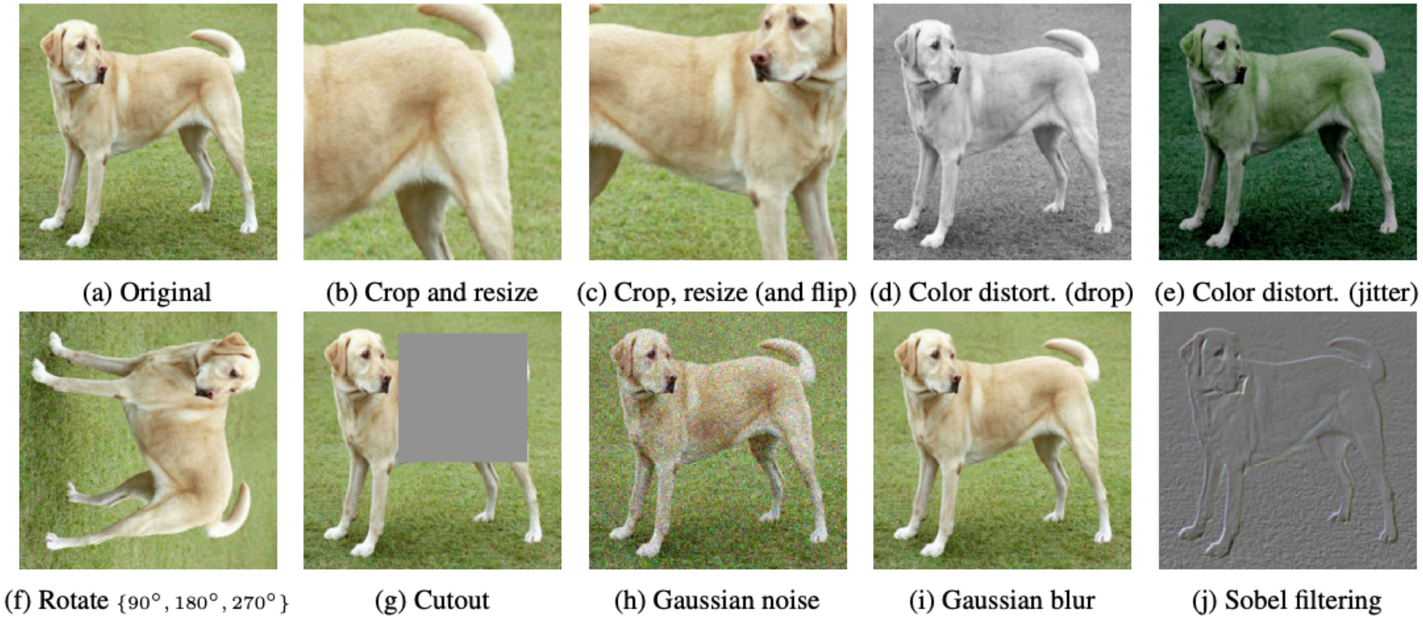


Figure 4. Illustrations of the studied data augmentation operators. Each augmentation can transform data stochastically with some internal parameters (e.g. rotation degree, noise level). Note that we *only* test these operators in ablation, the *augmentation policy used to train our models* only includes *random crop (with flip and resize)*, *color distortion*, and *Gaussian blur*. (Original image cc-by: Von.grzanka)

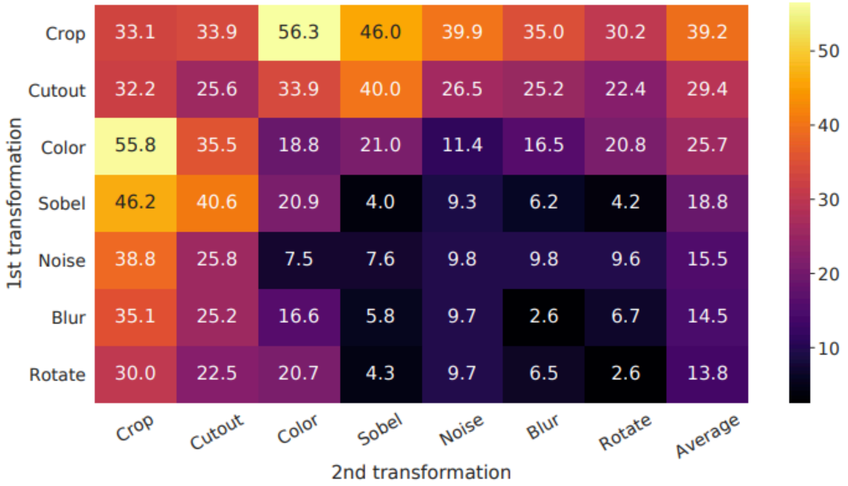
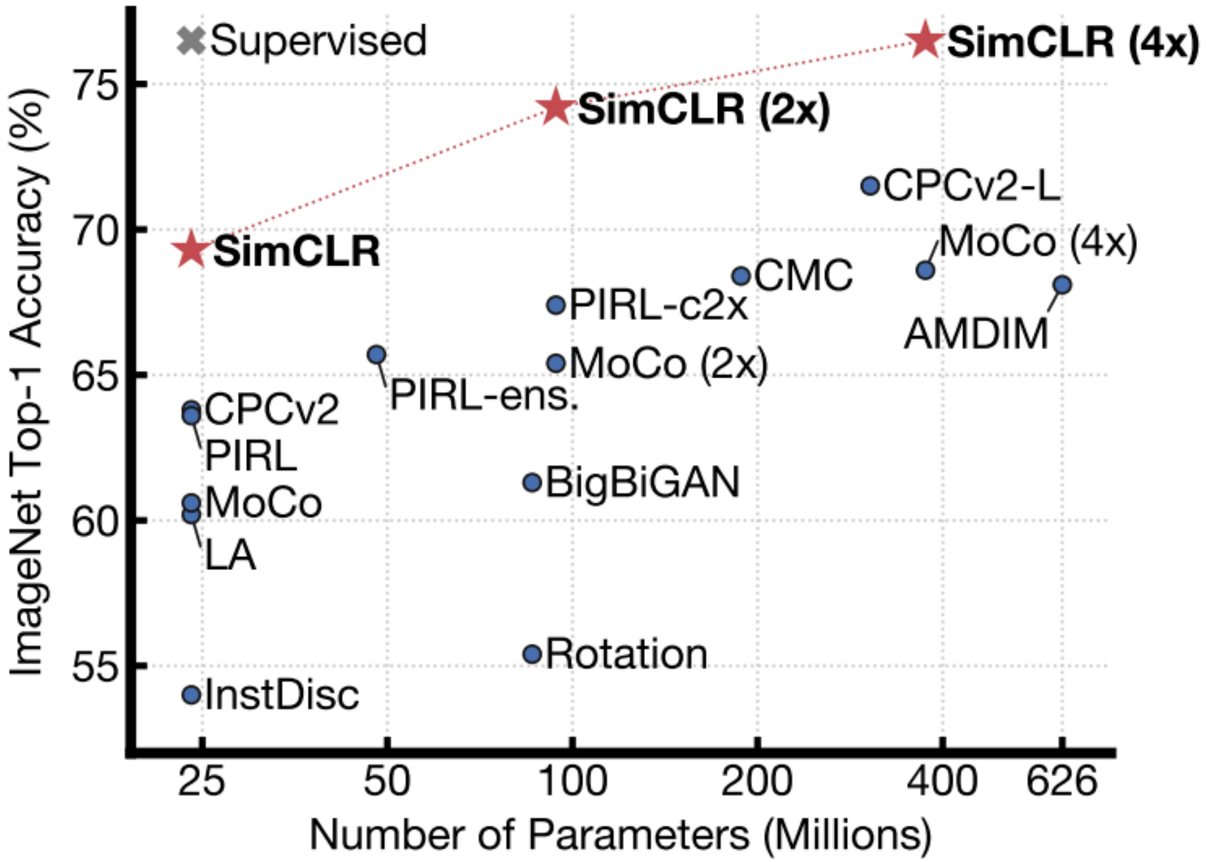


Figure 5. Linear evaluation (ImageNet top-1 accuracy) under individual or composition of data augmentations, applied only to one branch. For all columns but the last, diagonal entries correspond to single transformation, and off-diagonals correspond to composition of two transformations (applied sequentially). The last column reflects the average over the row.

T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. **A Simple Framework for Contrastive Learning of Visual Representations**. ICML 2020

SimCLR: Evaluation



No detection evaluation

T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. **A Simple Framework for Contrastive Learning of Visual Representations**. ICML 2020

Improved Baselines with Momentum Contrastive Learning

Xinlei Chen Haoqi Fan Ross Girshick Kaiming He
Facebook AI Research (FAIR)

Abstract

Contrastive unsupervised learning has recently shown encouraging progress, e.g., in Momentum Contrast (MoCo) and SimCLR. In this note, we verify the effectiveness of two of SimCLR's design improvements by implementing them in the MoCo framework. With simple modifications to MoCo—namely, using an MLP projection head and more data augmentation—we establish stronger baselines that outperform SimCLR and do not require large training batches. We hope this will make state-of-the-art unsupervised learning research more accessible. Code will be made public.

1. Introduction

Recent studies on unsupervised representation learning from images [16, 13, 8, 17, 1, 9, 15, 6, 12, 2] are converging on a central concept known as *contrastive learning* [5]. The

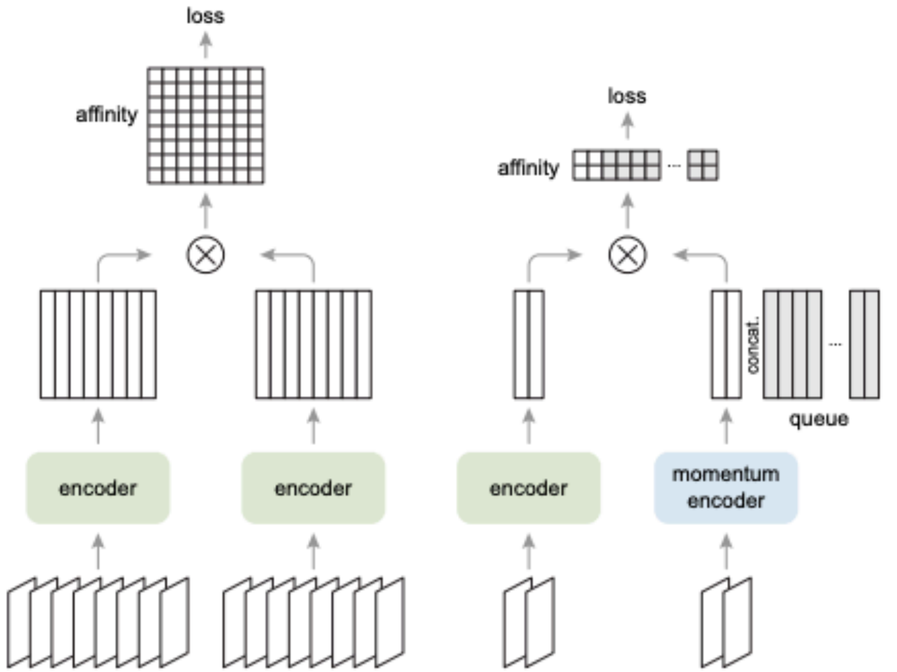


Figure 1. A **batching** perspective of two optimization mechanisms for contrastive learning. Images are encoded into a representation space, in which pairwise affinities are computed.

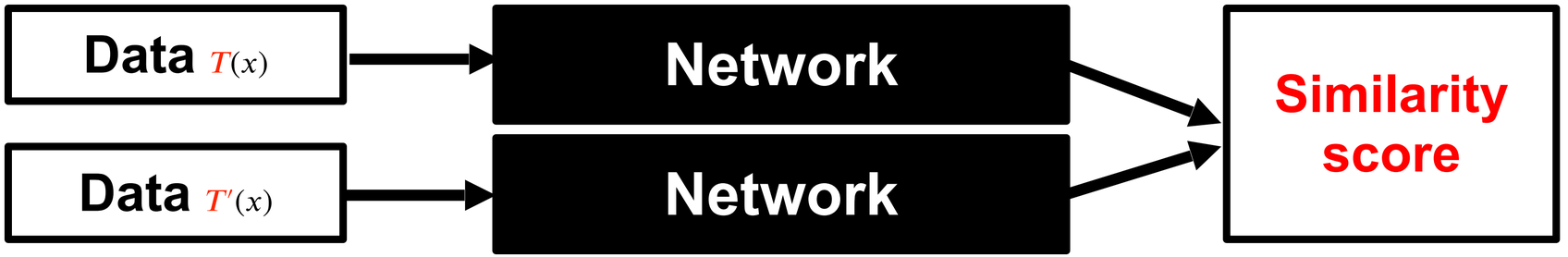
Ideas from SimCLR improve MoCo too!

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).

Non-contrastive methods

- Extract representations from two transformed versions of a data point, encourage these representations to be similar (or to have other desirable properties)
  - **Contrastive methods**: train using both positive (similar) and negative (dissimilar) pairs
    - Key challenge: sampling of negative pairs
  - **Non-contrastive methods**: train with only positive examples
    - Key challenge: avoiding degenerate solutions (all representations collapsing to constant output value)



Source: A. Efros

BYOL

- Use momentum encoder, but without the queue of negative examples
- Use projection head like SimCLR, add prediction head to online network

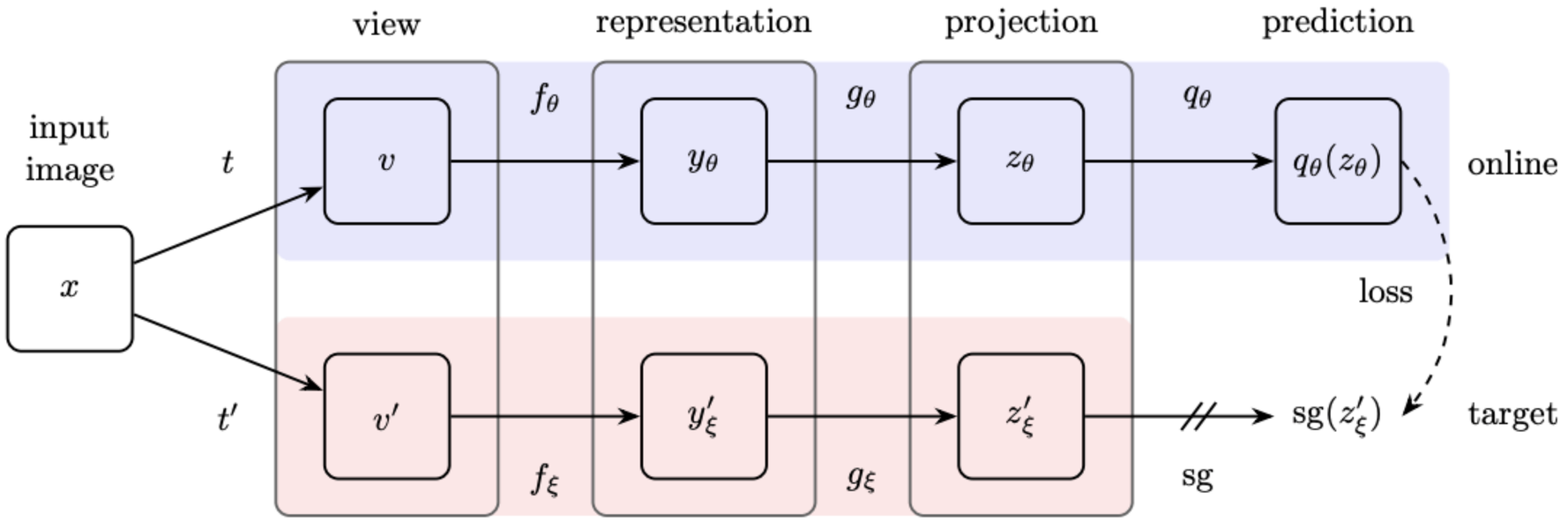


Figure 2: BYOL's architecture. BYOL minimizes a similarity loss between $q_\theta(z_\theta)$ and $\mathrm{sg}(z'_\xi)$, where θ are the trained weights, ξ are an exponential moving average of θ and sg means stop-gradient. At the end of training, everything but f_θ is discarded, and y_θ is used as the image representation.

J.-B. Grill et al. **Bootstrap Your Own Latent A New Approach to Self-Supervised Learning**. NeurIPS 2020

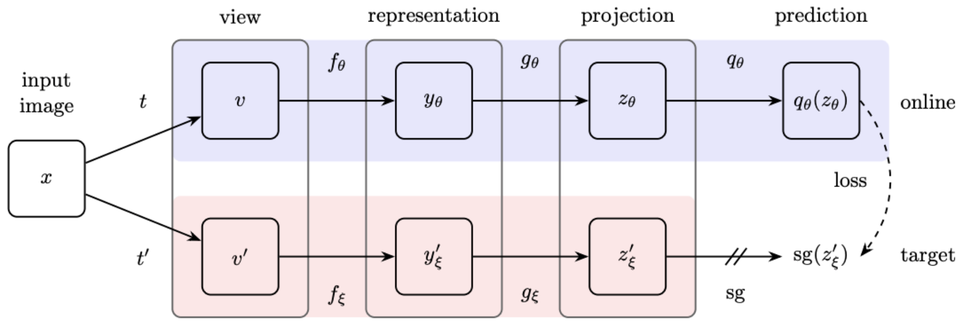
BYOL

Minimize squared error between normalized predictions and target projections:

$$\mathcal{L}_{\theta,\xi} \triangleq \left\| \overline{q_\theta}(z_\theta) - \overline{z}'_\xi \right\|_2^2 = 2 - 2 \cdot \frac{\langle q_\theta(z_\theta), z'_\xi \rangle}{\left\| q_\theta(z_\theta) \right\|_2 \cdot \left\| z'_\xi \right\|_2}.$$

Update parameters:

$$\theta \leftarrow \text{optimizer}\left(\theta, \nabla_\theta \mathcal{L}_{\theta,\xi}^{\texttt{BYOL}}, \eta\right),$$
$$\xi \leftarrow \tau\xi + (1-\tau)\theta,$$



Keep f_θ at the end

BYOL: Evaluation

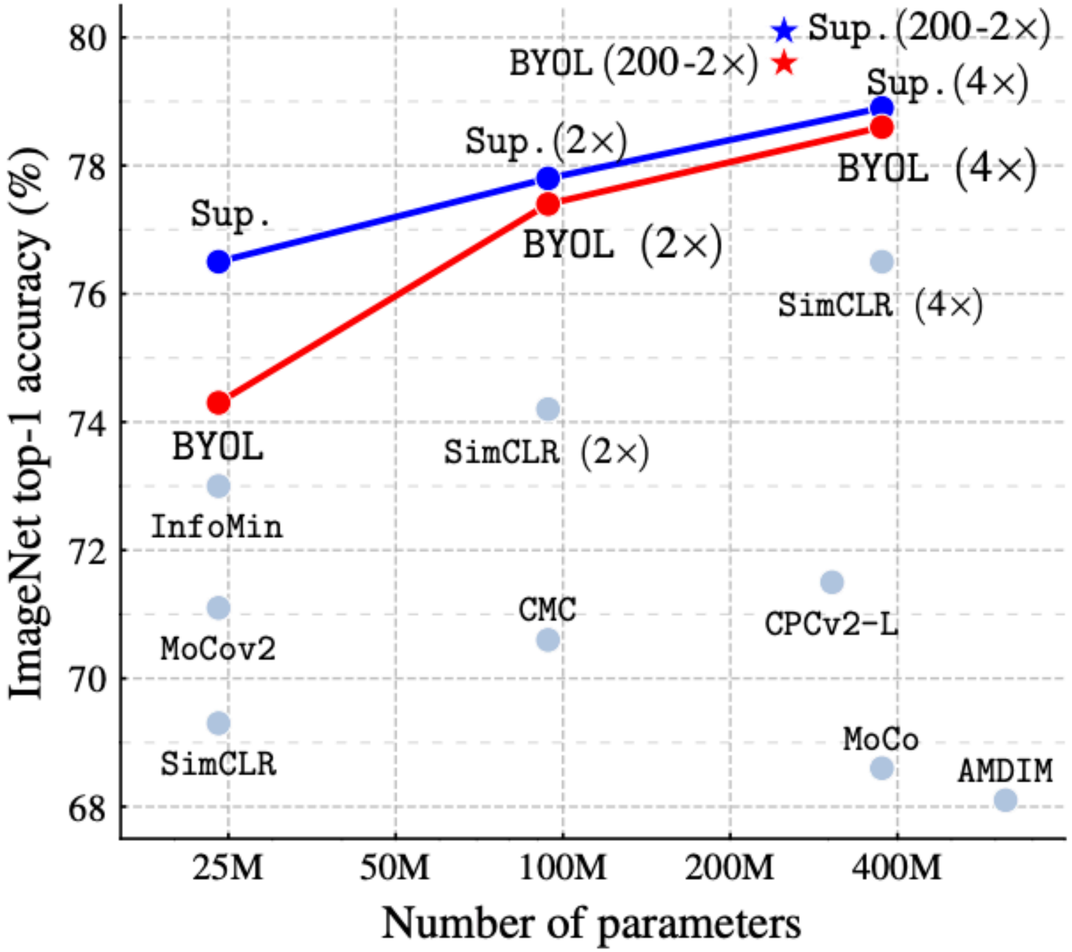


Figure 1: Performance of BYOL on ImageNet (linear evaluation) using ResNet-50 and our best architecture ResNet-200 (2×), compared to other unsupervised and supervised (Sup.) baselines [8].

Why does BYOL work?

Why is does this lead to a non-trivial solution?

A trivial solution is to map every image to a constant vector!

Asymmetric learning — two networks have different learning rules, one uses SGD on the loss and other uses EMA

Does batch normalization play a role?

- Removing BN resulted in a network that performed no better than random. Originally a bug in the code!
- https://imbue.com/research/2020-08-24-understanding-self-supervised-contrastive-learning/

DINO

Similar to BYOL

Softmax labels + cross-entropy loss

Extensive use of ViTs

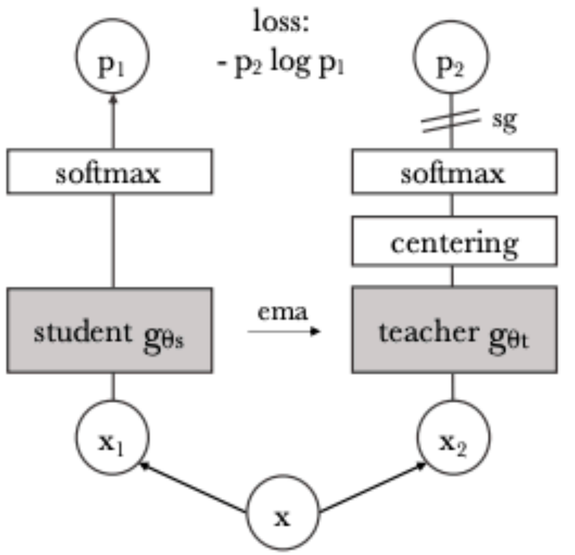


Figure 2: **Self-distillation with no labels.** We illustrate DINO in the case of one single pair of views (x_1, x_2) for simplicity. The model passes two different random transformations of an input image to the student and teacher networks. Both networks have the same architecture but different parameters. The output of the teacher network is centered with a mean computed over the batch. Each networks outputs a K dimensional feature that is normalized with a temperature softmax over the feature dimension. Their similarity is then measured with a cross-entropy loss. We apply a stop-gradient (sg) operator on the teacher to propagate gradients only through the student. The teacher parameters are updated with an exponential moving average (ema) of the student parameters.

Emerging Properties in Self-Supervised Vision Transformers, Caron et al., 2021

DINO

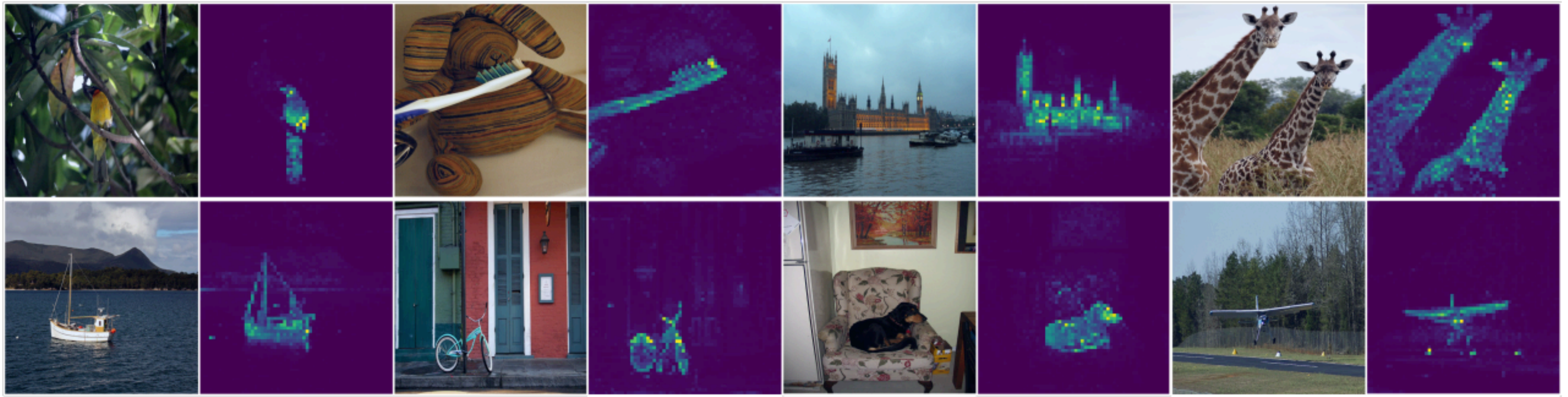


Figure 1: **Self-attention from a Vision Transformer with 8×8 patches trained with no supervision.** We look at the self-attention of the `[CLS]` token on the heads of the last layer. This token is not attached to any label nor supervision. These maps show that the model automatically learns class-specific features leading to unsupervised object segmentations.

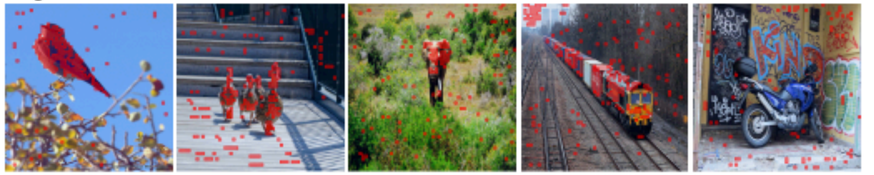
Emerging Properties in Self-Supervised Vision Transformers, Caron et al., 2021

DINO

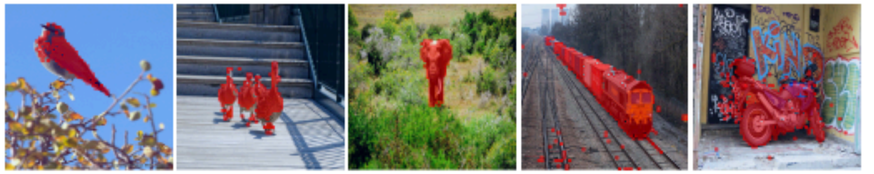
Table 2: **Linear and k-NN classification on ImageNet.** We report top-1 accuracy for linear and k-NN evaluations on the validation set of ImageNet for different self-supervised methods. We focus on ResNet-50 and ViT-small architectures, but also report the best results obtained across architectures. * are run by us. We run the k-NN evaluation for models with official released weights. The throughput (im/s) is calculated on a NVIDIA V100 GPU with 128 samples per forward. Parameters (M) are of the feature extractor.

Method	Arch.	Param.	im/s	Linear	k-NN
Supervised	RN50	23	1237	79.3	79.3
SCLR [12]	RN50	23	1237	69.1	60.7
MoCov2 [15]	RN50	23	1237	71.1	61.9
InfoMin [67]	RN50	23	1237	73.0	65.3
BarlowT [81]	RN50	23	1237	73.2	66.0
OBoW [27]	RN50	23	1237	73.8	61.9
BYOL [30]	RN50	23	1237	74.4	64.8
DCv2 [10]	RN50	23	1237	75.2	67.1
SwAV [10]	RN50	23	1237	**75.3**	65.7
DINO	RN50	23	1237	**75.3**	**67.5**
Supervised	ViT-S	21	1007	79.8	79.8
BYOL* [30]	ViT-S	21	1007	71.4	66.6
MoCov2* [15]	ViT-S	21	1007	72.7	64.4
SwAV* [10]	ViT-S	21	1007	73.5	66.3
DINO	ViT-S	21	1007	**77.0**	**74.5**

Supervised



DINO



	Random	Supervised	DINO
ViT-S/16	22.0	27.3	45.9
ViT-S/8	21.8	23.7	44.7

Figure 4: **Segmentations from supervised versus DINO.** We visualize masks obtained by thresholding the self-attention maps to keep 60% of the mass. On top, we show the resulting masks for a ViT-S/8 trained with supervision and DINO. We show the best head for both models. The table at the bottom compares the Jaccard similarity between the ground truth and these masks on the validation images of PASCAL VOC12 dataset.

Also good at spatial tasks

Emerging Properties in Self-Supervised Vision Transformers, Caron et al., 2021

Post 2021

Transition from ConvNets to Vision Transformers

Scaling SSL to larger datasets and model sizes

Improved performance on spatial tasks (detection, segmentation, image matching)

Masked auto-encoders

Inspired by masked auto-encoders from language

Works especially well for ViTs as they operate on patches

The [MASK] of France is [MASK].

capital Paris

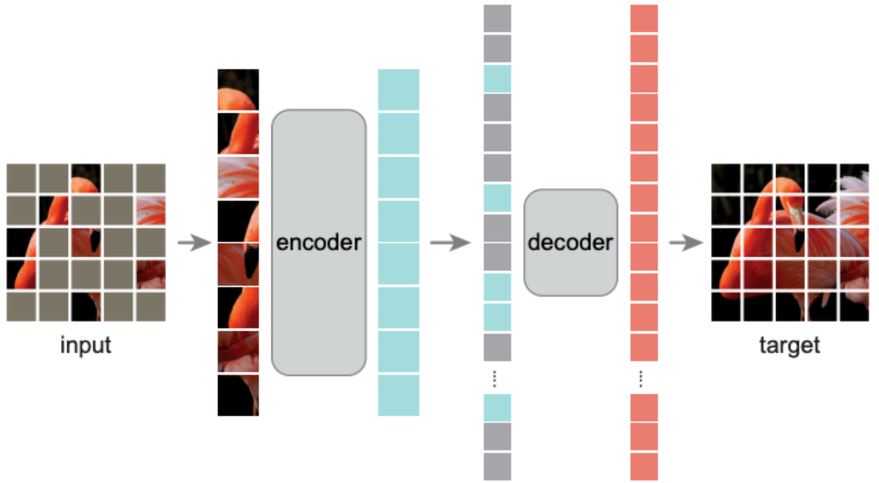


Figure 1. **Our MAE architecture**. During pre-training, a large random subset of image patches (*e.g.*, 75%) is masked out. The encoder is applied to the small subset of *visible patches*. Mask tokens are introduced *after* the encoder, and the full set of encoded patches and mask tokens is processed by a small decoder that reconstructs the original image in pixels. After pre-training, the decoder is discarded and the encoder is applied to uncorrupted images (full sets of patches) for recognition tasks.

Masked Autoencoders Are Scalable Vision Learners, He et al., 2021

Masked auto-encoders

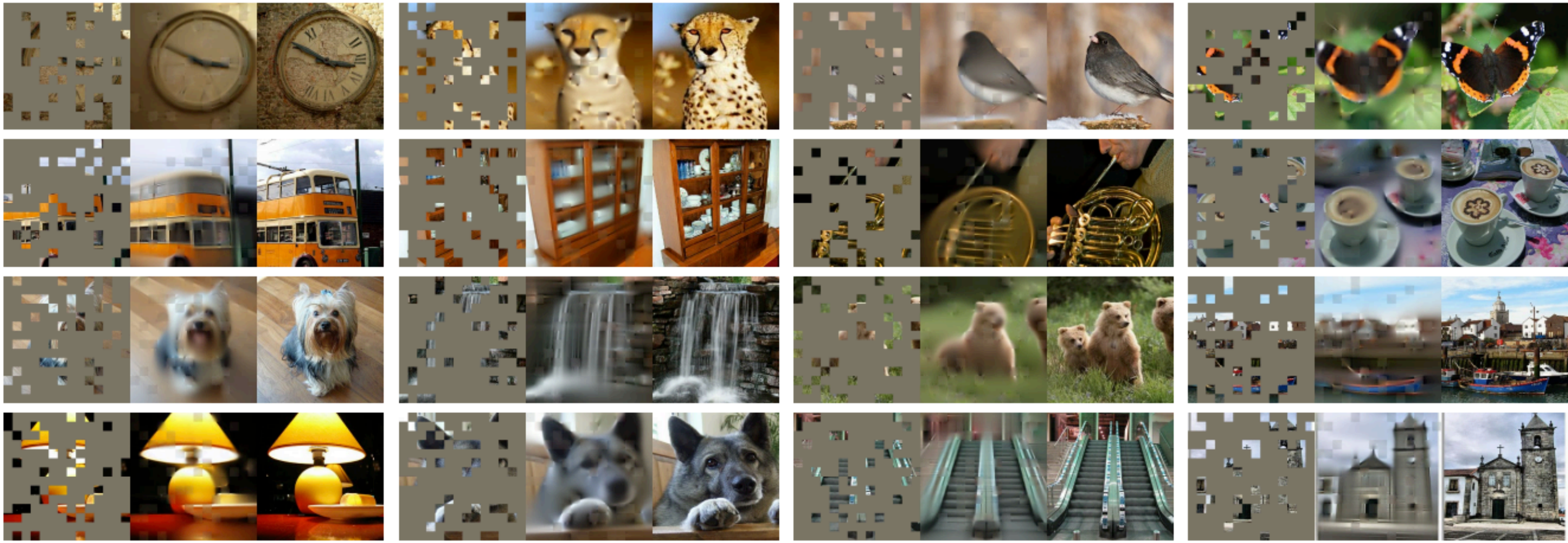


Figure 2. Example results on ImageNet *validation* images. For each triplet, we show the masked image (left), our MAE reconstruction (middle), and the ground-truth (right). The masking ratio is 80%, leaving only 39 out of 196 patches. More examples are in the appendix

†*As no loss is computed on visible patches, the model output on visible patches is qualitatively worse. One can simply overlay the output with the visible patches to improve visual quality. We intentionally opt not to do this, so we can more comprehensively demonstrate the method's behavior.*

Masked Autoencoders Are Scalable Vision Learners, He et al., 2021

Masked auto-encoders

method	pre-train data	ViT-B	ViT-L	ViT-H	ViT-H_{448}
scratch, our impl.	-	82.3	82.6	83.1	-
DINO [5]	IN1K	82.8	-	-	-
MoCo v3 [9]	IN1K	83.2	84.1	-	-
BEiT [2]	IN1K+DALLE	83.2	85.2	-	-
MAE	IN1K	83.6	85.9	86.9	**87.8**

Table 3. **Comparisons with previous results on ImageNet-1K**. The pre-training data is the ImageNet-1K training set (except the tokenizer in BEiT was pre-trained on 250M DALLE data [50]). All self-supervised methods are evaluated by end-to-end fine-tuning. The ViT models are B/16, L/16, H/14 [16]. The best for each column is underlined. All results are on an image size of 224, except for ViT-H with an extra result on 448. Here our MAE reconstructs normalized pixels and is pre-trained for 1600 epochs.

		AP^{box}		AP^{mask}	
method	pre-train data	ViT-B	ViT-L	ViT-B	ViT-L
supervised	IN1K w/ labels	47.9	49.3	42.9	43.9
MoCo v3	IN1K	47.9	49.3	42.7	44.0
BEiT	IN1K+DALLE	49.8	**53.3**	44.4	47.1
MAE	IN1K	**50.3**	**53.3**	**44.9**	**47.2**

Table 4. **COCO object detection and segmentation** using a ViT Mask R-CNN baseline. All entries are based on our implementation. Self-supervised entries use IN1K data *without* labels. Mask AP follows a similar trend as box AP.

Starting to beat supervised ImageNet, but requires lot more compute!

Masked Autoencoders Are Scalable Vision Learners, He et al., 2021

DINOv3

Careful model and dataset scaling

Model sizes up to 7B parameters compared to ~1B for DINO

Training set ~ 17 billion images subsampled to 1,689 million images though various data curation strategies

Training objective: DINO loss + … + Gram-Anchoring loss

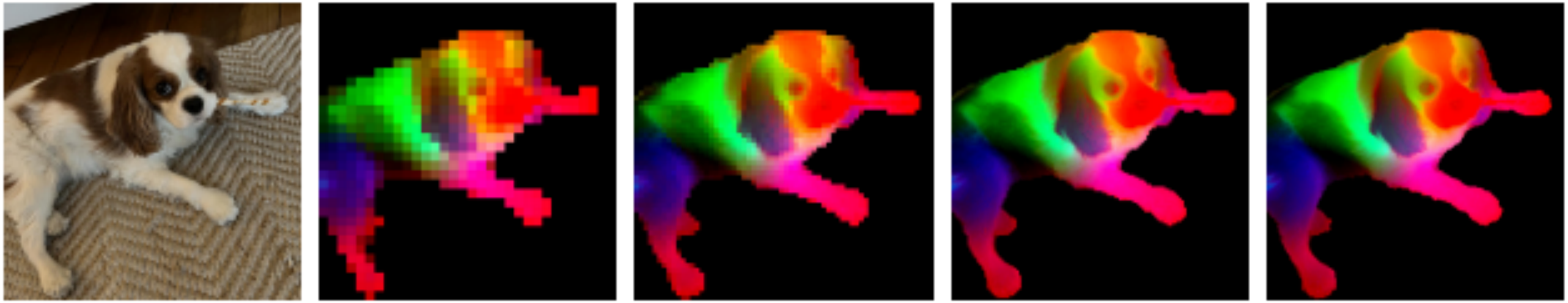
Gram-Anchoring encourages attention maps across layers to be similar — fights the tendency of ViTs to collapse semantics into specific tokens!

Exceptional performance on spatial tasks

DINOv3

Figure 3: High-resolution dense features. We visualize the cosine similarity maps obtained with DINOv3 output features between the patches marked with a red cross and all other patches. Input image at 4096×4096. *Please zoom in*, do you agree with DINOv3?

DINOv3



Input | 512 × 512 | 1024 × 1024 | 2048 × 2048 | 4096 × 4096

Figure 4: DINOv3 at very high resolution. We visualize dense features of DINOv3 by mapping the first three components of a PCA computed over the feature space to RGB. To focus the PCA on the subject, we mask the feature maps via background subtraction. With increasing resolution, DINOv3 produces crisp features that stay semantically meaningful. We visualize more PCAs in Sec. 6.1.1.

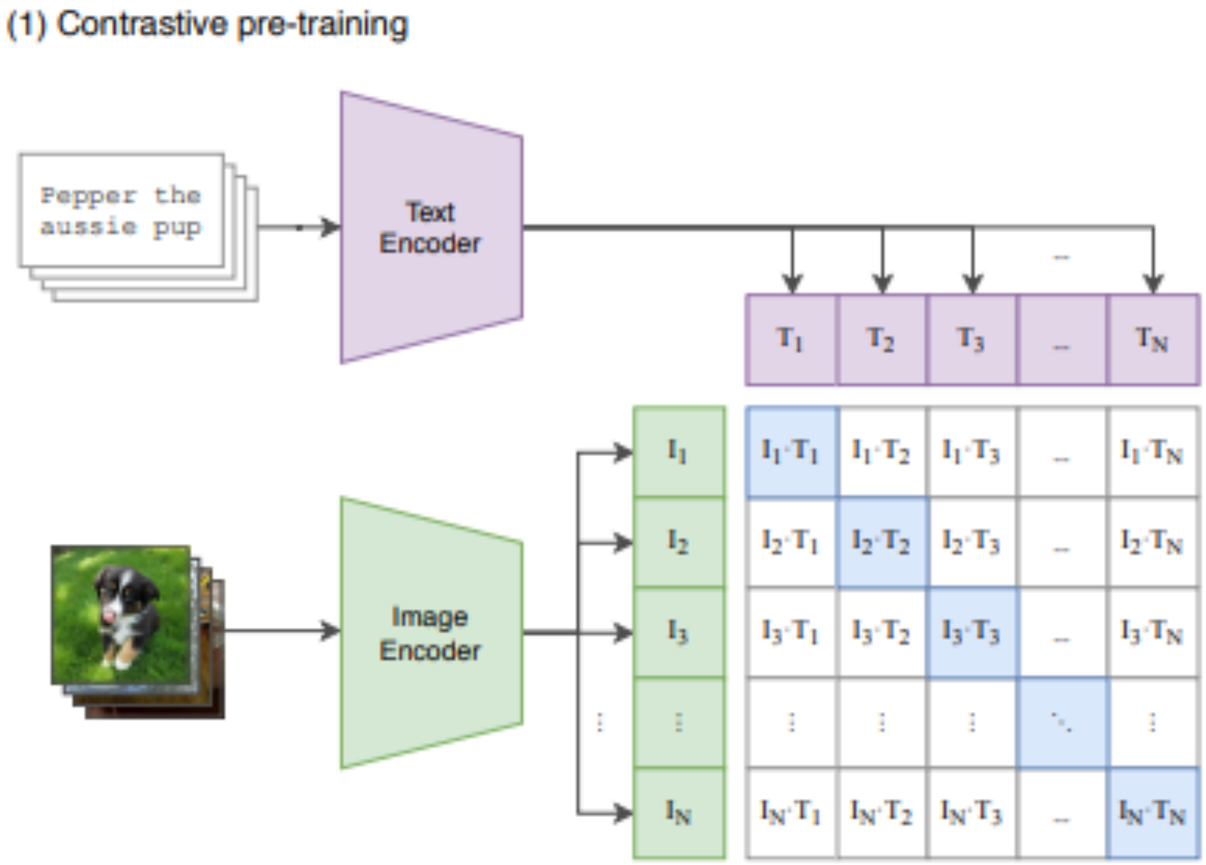
Self-supervised learning: Outline

- Data prediction
  - Colorization
- Transformation prediction
  - Context prediction, jigsaw puzzle solving, rotation prediction
- “Siamese” methods
  - Contrastive methods
  - Non-contrastive methods
- Self-supervision beyond still images
  - Video, audio, language

CLIP: Connecting text and images

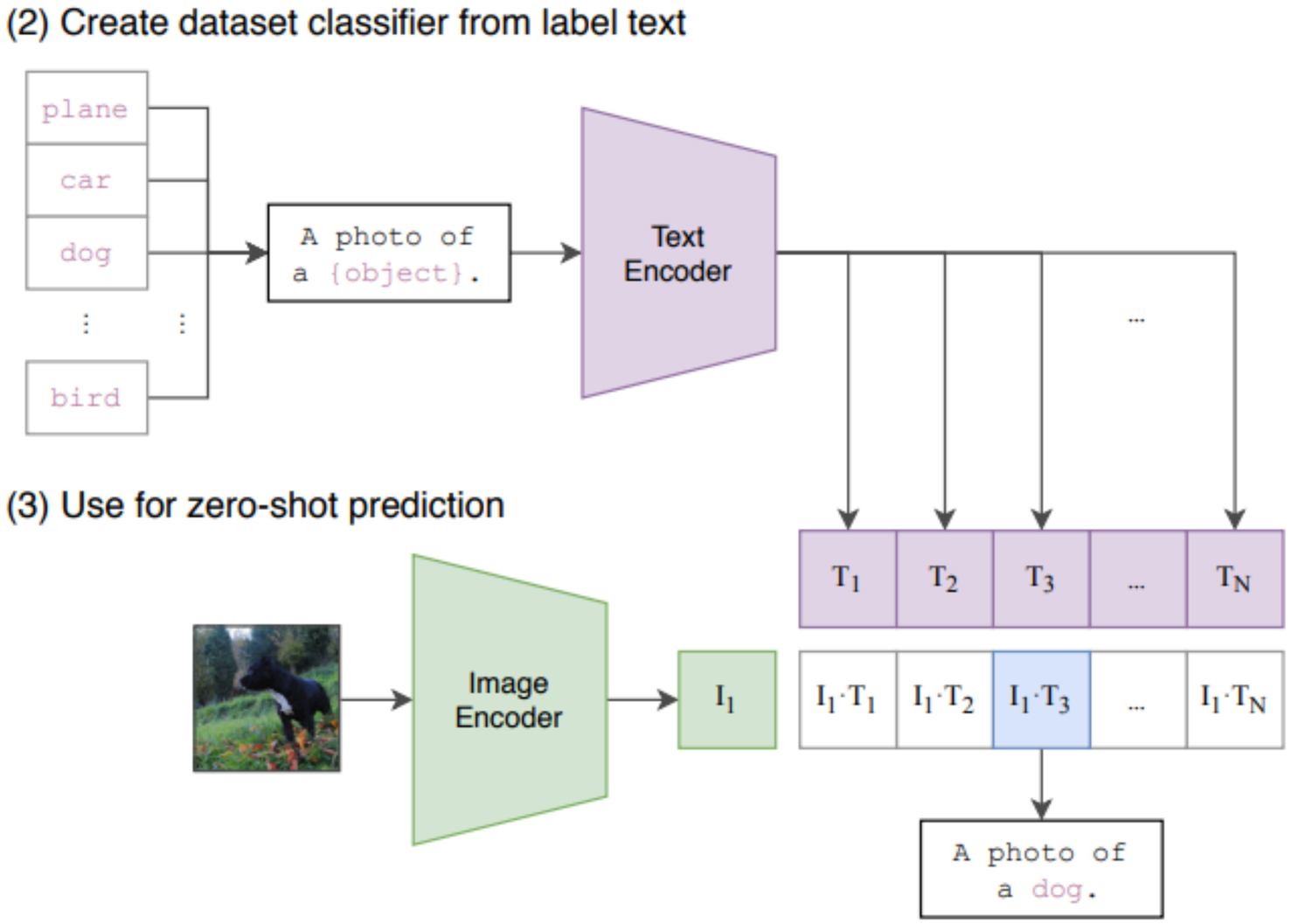
Trained on 400 million (image, text) pairs

Models and code available on “OpenAI CLIP”



CLIP: Connecting text and images

Zero-shot prediction via text prompts



CLIP: Connecting text and images

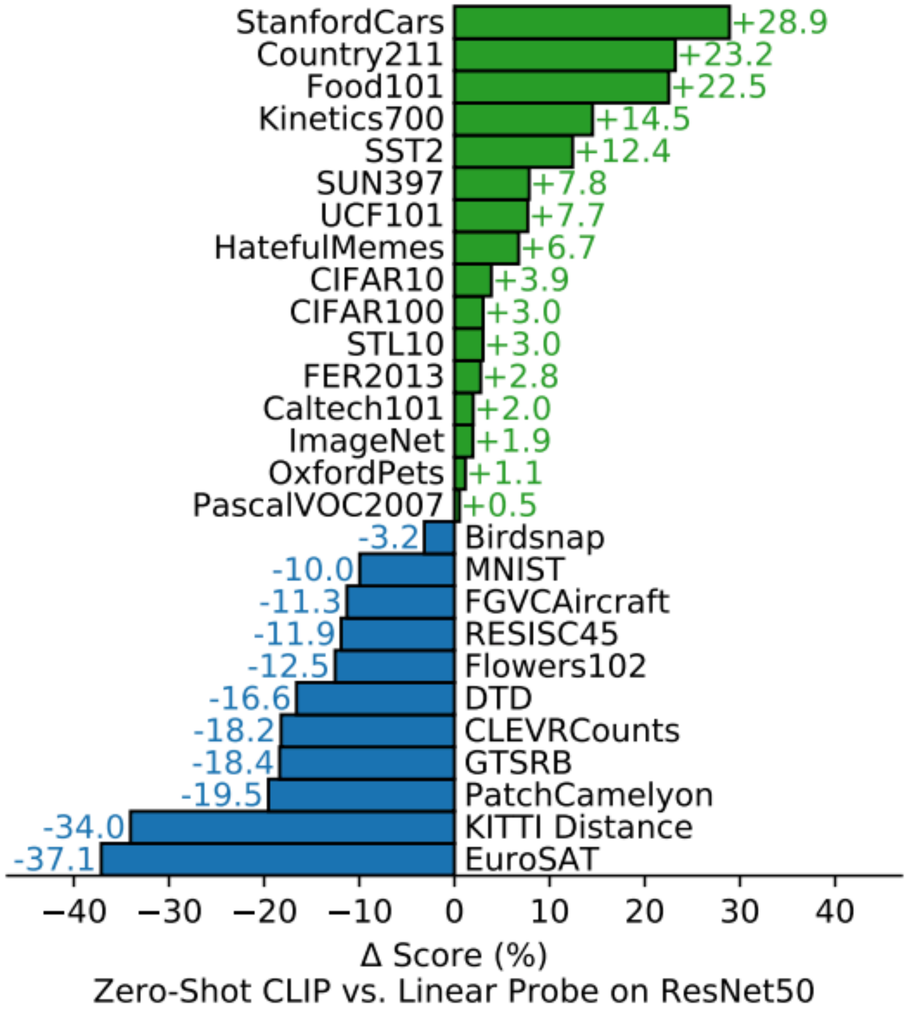


Figure 5. **Zero-shot CLIP is competitive with a fully supervised baseline.** Across a 27 dataset eval suite, a zero-shot CLIP classifier outperforms a fully supervised linear classifier fitted on ResNet-50 features on 16 datasets, including ImageNet.

WildSAT — Learning Satellite Image Representations

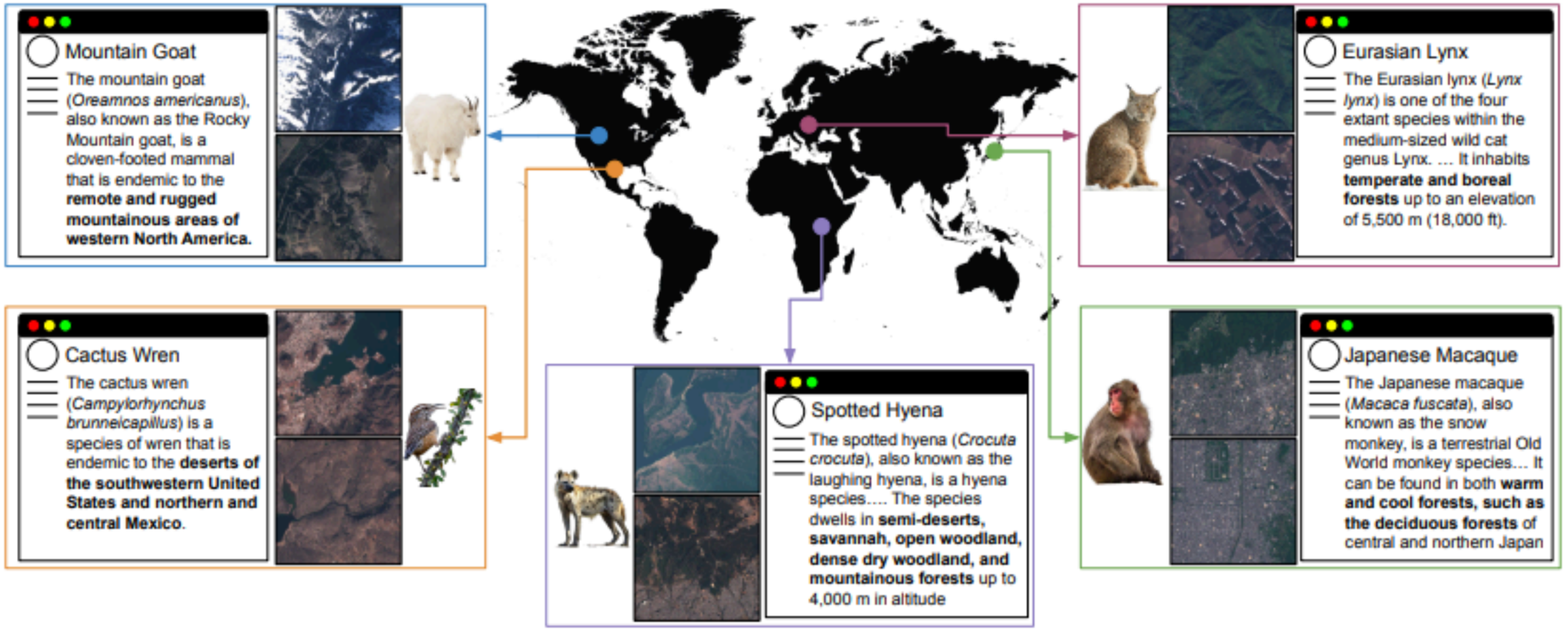


Figure 1. **Wildlife observations can provide valuable supervision for learning satellite image representations.** Known wildlife locations derived from human observations, coupled with descriptive information on species range, habitat, and other ecological attributes on Wikipedia, serve as a rich source of contextual information for satellite imagery. Our **WildSAT** approach leverages these additional data sources to (i) learn robust satellite image representations for downstream tasks, and (ii) complement and further improve existing models using continual pre-training.

WildSAT: Learning Satellite Image Representations from Wildlife Observations, Rangel Daroya, Elijah Cole, Oisin Mac Aodha, Grant Van Horn, Subhransu Maji, ICCV'25

WildSAT — Learning Satellite Image Representations

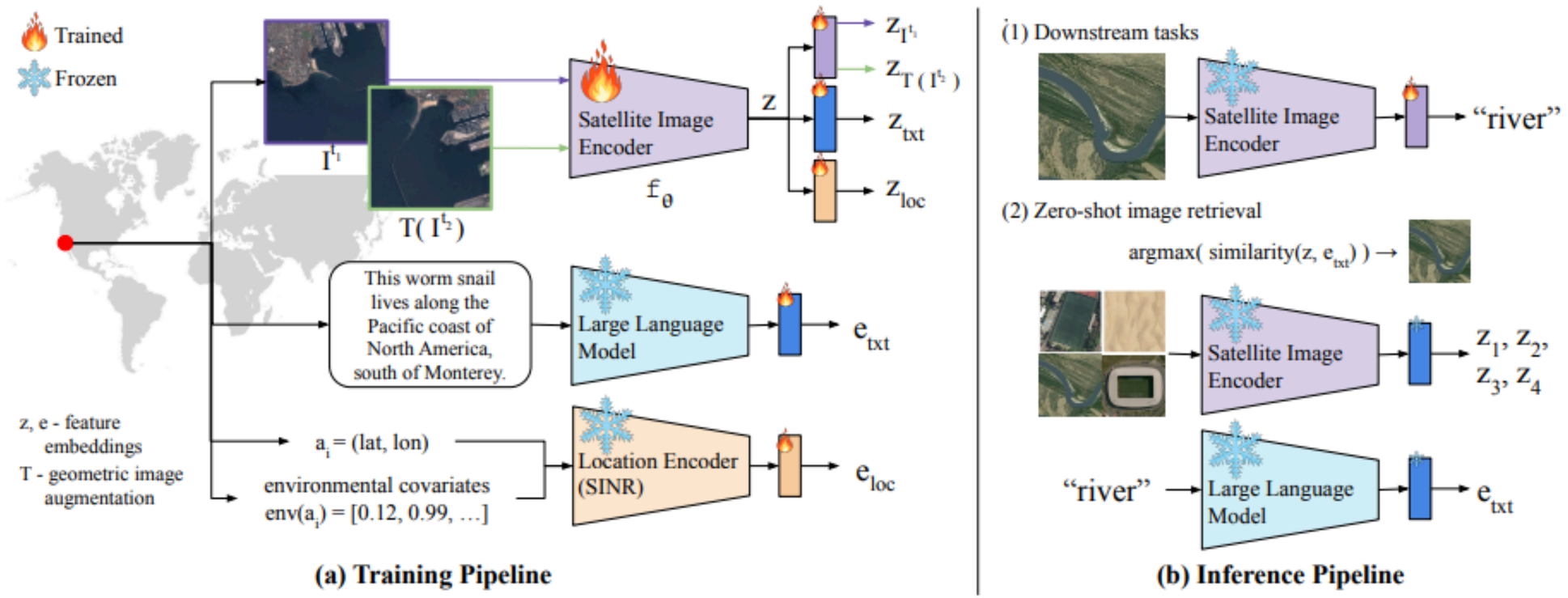
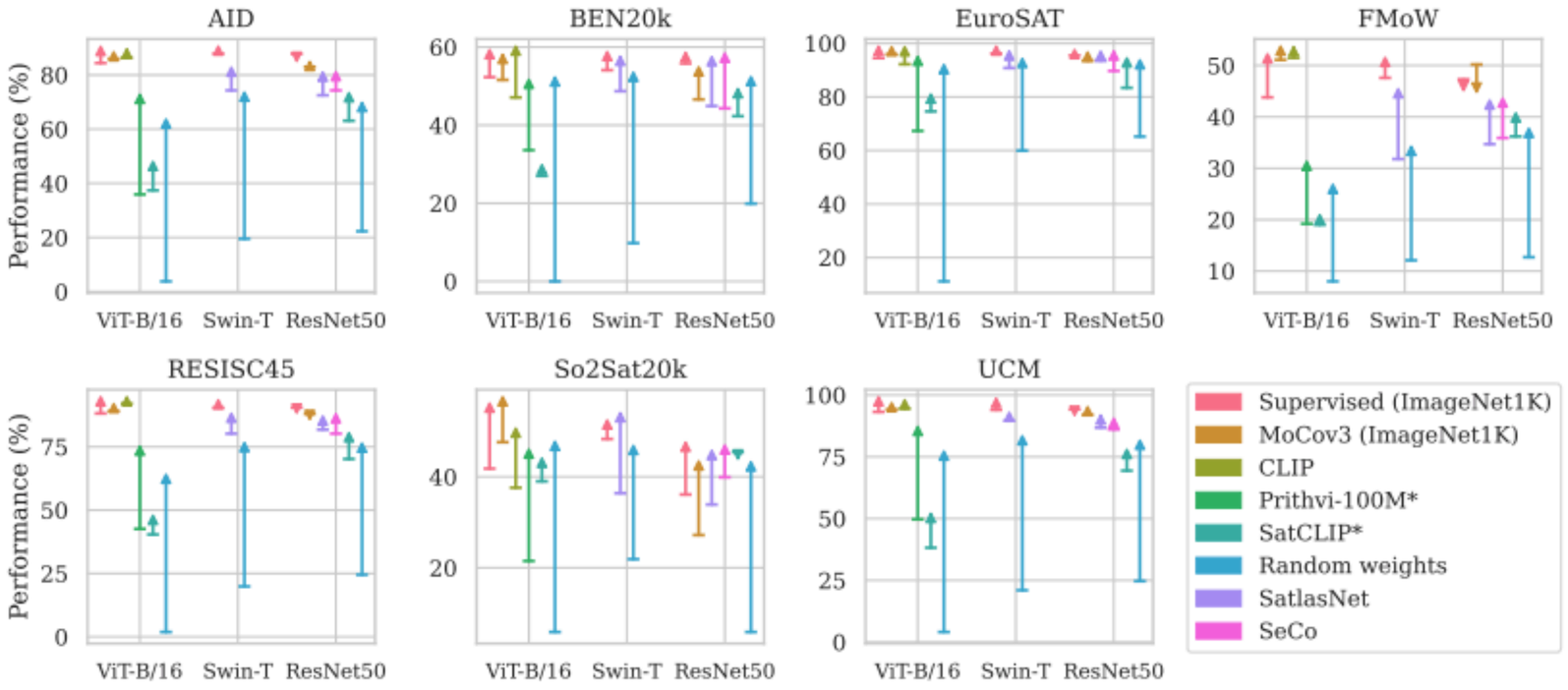


Figure 2. **Architecture for training and evaluating the satellite image encoder**. (a) The training pipeline uses the location of a species, the satellite images at those locations, the environmental covariates, and the Wikipedia text associated with the species. In addition to the alignment of image, text, and location modalities, the encoder is encouraged to learn additional image features by using temporal and geometric image transformations on the input satellite image. (b) Downstream tasks use the frozen satellite image encoder with an additional trainable layer (or layers). Alternatively, the predicted image embeddings can be used for zero-shot retrieval via text queries.

WildSAT: Learning Satellite Image Representations from Wildlife Observations, Rangel Daroya, Elijah Cole, Oisin Mac Aodha, Grant Van Horn, Subhransu Maji, ICCV'25

WildSAT — Learning Satellite Image Representations



Dataset	Average (w/ random) Base	+WS
AID [76]	61.2	**77.0**
BEN20k [38, 65]	38.5	**53.1**
EuroSAT [28]	80.2	**93.8**
FMoW [9]	33.4	**41.1**
RESISC45 [8]	65.3	**81.0**
So2Sat20k [38, 84]	32.6	**47.6**
UCM [78]	68.8	**86.1**

Dataset	Average (no random) Base	+WS
AID [76]	72.7	**79.4**
BEN20k [38, 65]	45.7	**53.4**
EuroSAT [28]	88.9	**94.3**
FMoW [9]	39.0	**43.3**
RESISC45 [8]	77.8	**83.5**
So2Sat20k [38, 84]	37.9	**48.2**
UCM [78]	81.8	**87.9**

Figure 3. **Linear probing performance improvement on seven downstream datasets without (Base) and with WildSAT (+WS) fine-tuning**. Accuracy is visualized for all dataset plots except BEN20k that visualizes micro F1 score. For each architecture and pre-training combination, the horizontal line marker represents the performance of the original model, while the triangle marker indicates performance after additional training on species observation data (WildSAT). The tables on the right summarize average performance across all seven datasets: the top table includes models with random weights, and the bottom table excludes them. Across the board, fine-tuning with species observation data leads to notable performance gains over most base models. We include the raw numbers in Tab. A1 (Appendix). *Both Prithvi-100M and SatCLIP are pre-trained with multispectral images, but for consistency across downstream datasets and models, only RGB bands are used here. We show that WildSAT also improves on multispectral images in Tab. A4 (Appendix).

WildSAT: Learning Satellite Image Representations from Wildlife Observations, Rangel Daroya, Elijah Cole, Oisin Mac Aodha, Grant Van Horn, Subhransu Maji, ICCV'25

WildSAT — Learning Satellite Image Representations

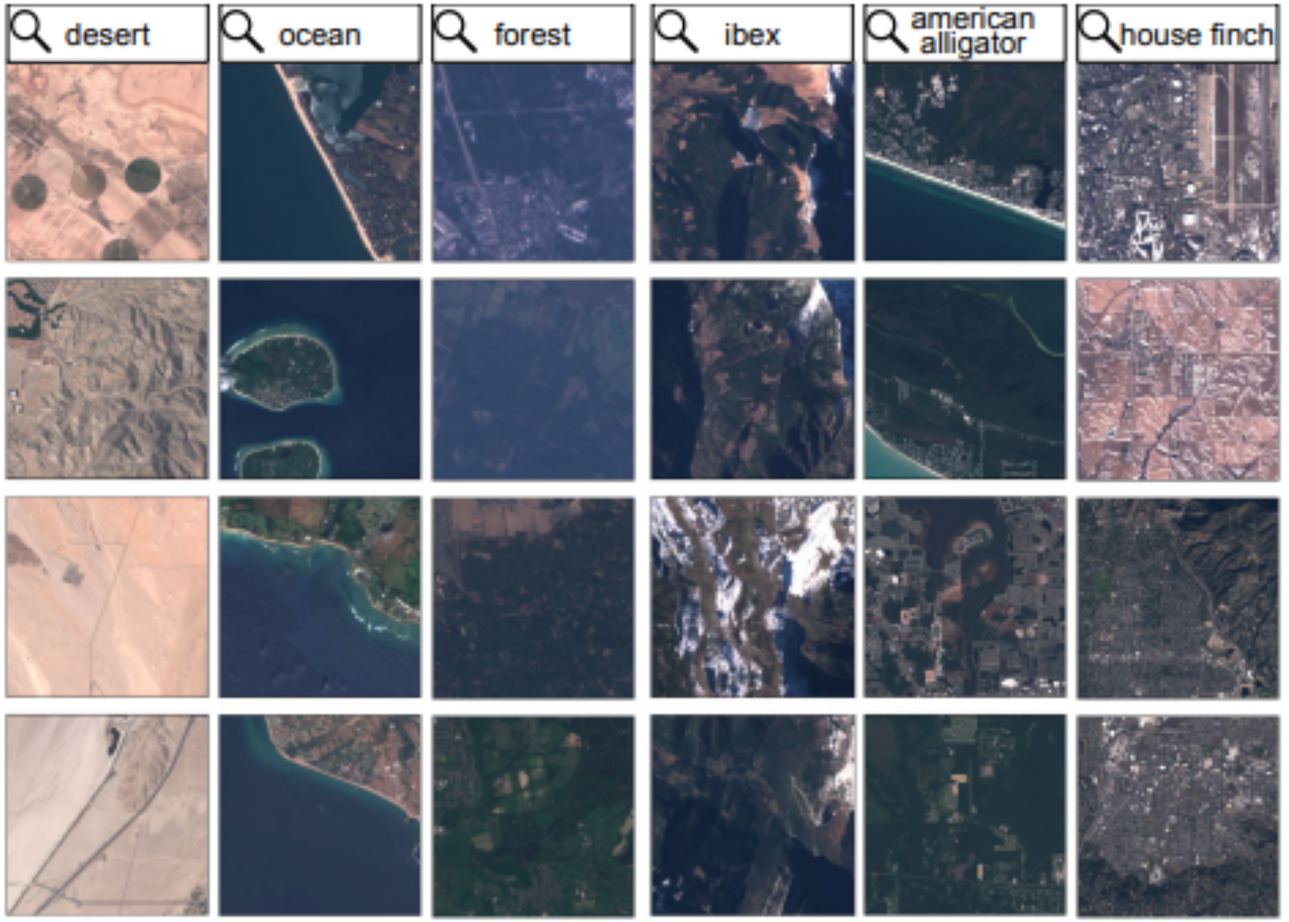


Figure 4. **Zero-shot results for text-based satellite image retrieval**. The columns show the top 4 images returned given the text query on top. A model can be queried using general landscape

WildSAT — Learning Satellite Image Representations

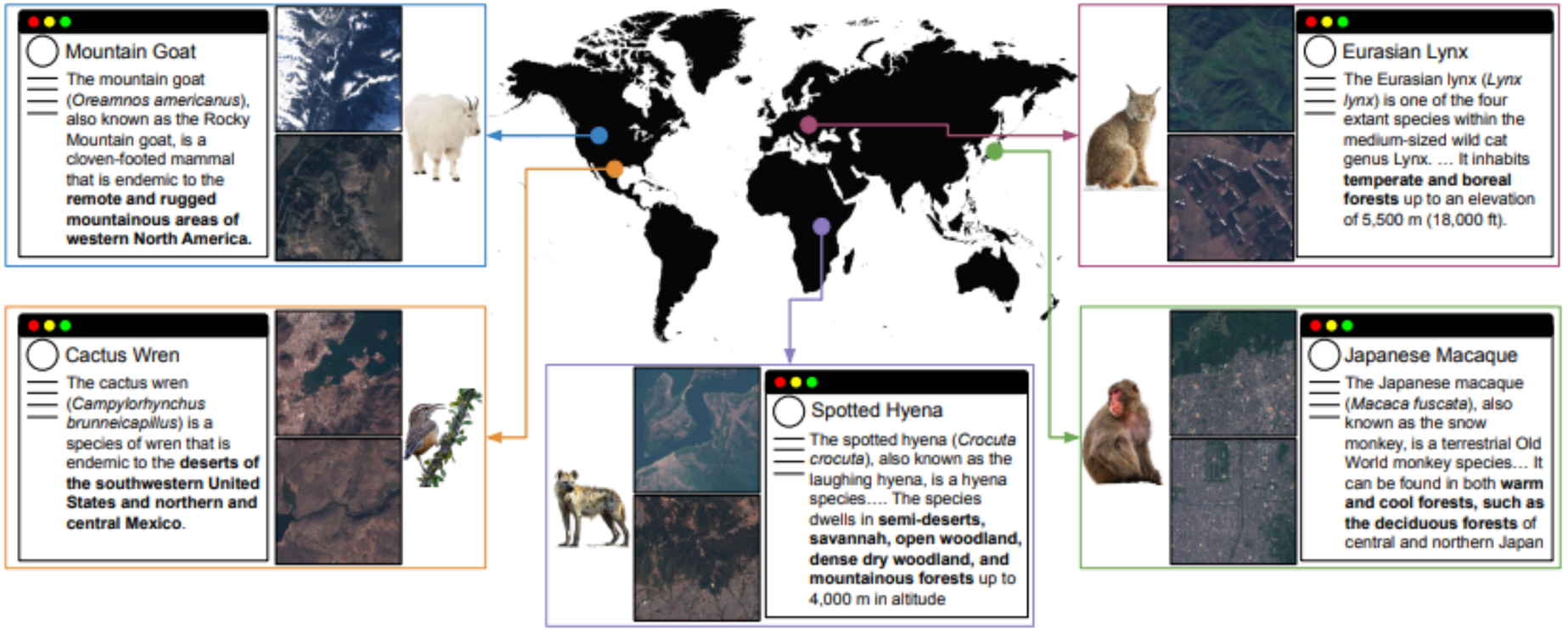
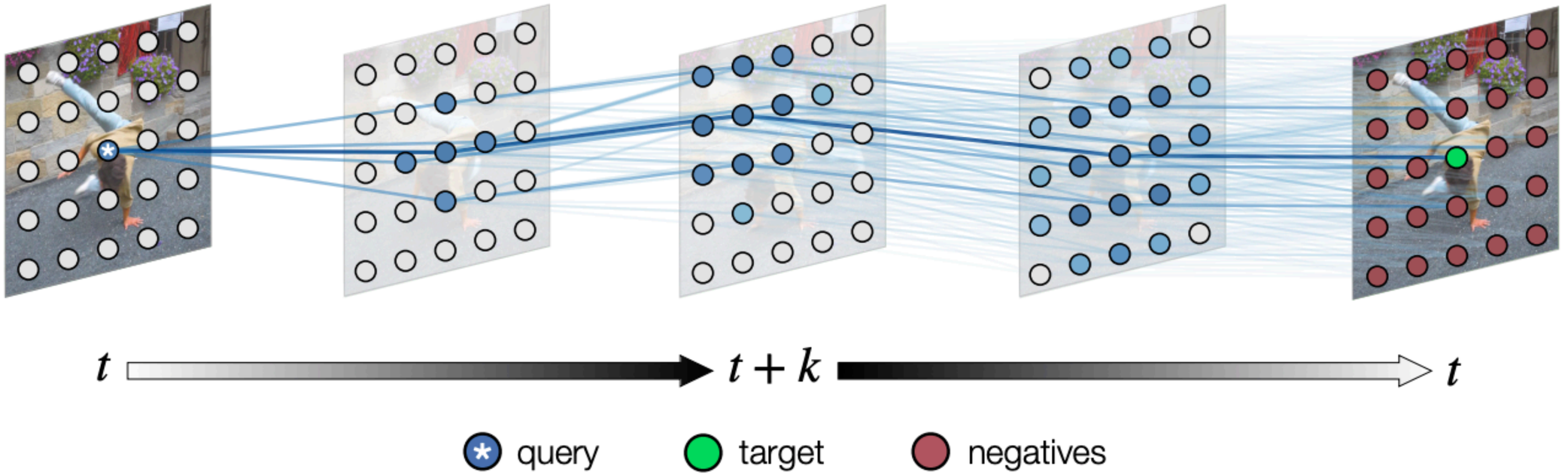


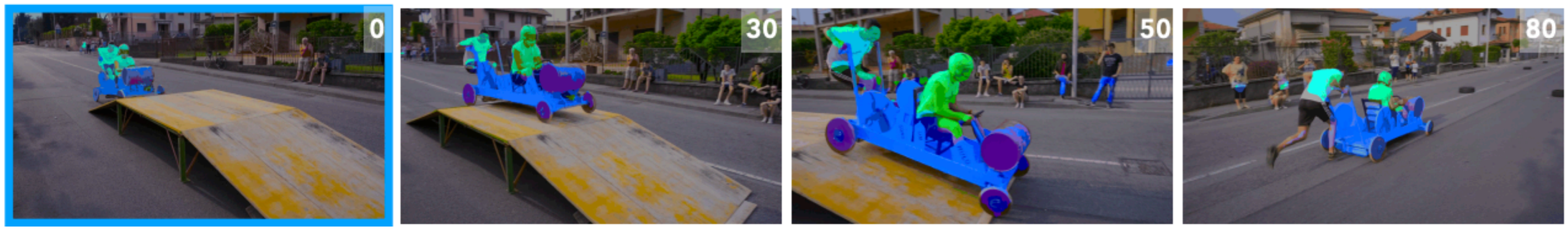
Figure 1. **Wildlife observations can provide valuable supervision for learning satellite image representations.** Known wildlife locations derived from human observations, coupled with descriptive information on species range, habitat, and other ecological attributes on Wikipedia, serve as a rich source of contextual information for satellite imagery. Our **WildSAT** approach leverages these additional data sources to (i) learn robust satellite image representations for downstream tasks, and (ii) complement and further improve existing models using continual pre-training.

WildSAT: Learning Satellite Image Representations from Wildlife Observations, Rangel Daroya, Elijah Cole, Oisin Mac Aodha, Grant Van Horn, Subhransu Maji, ICCV'25

Video correspondence features

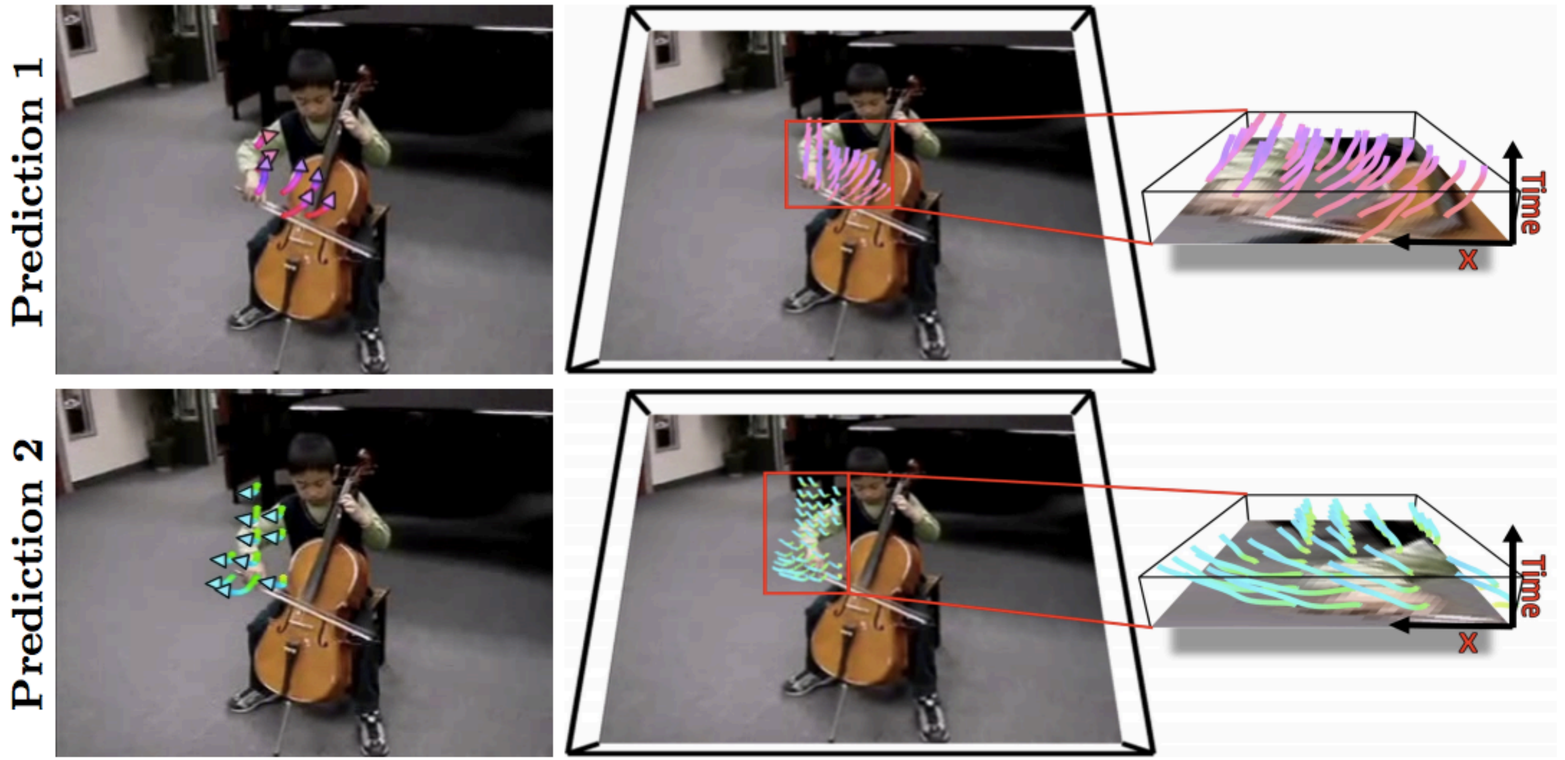


Object Propagation 1-4 Objects



A. Jabri, A. Owens, and A. Efros. **Space-time correspondence as a contrastive random walk**. NeurIPS 2020

Future prediction



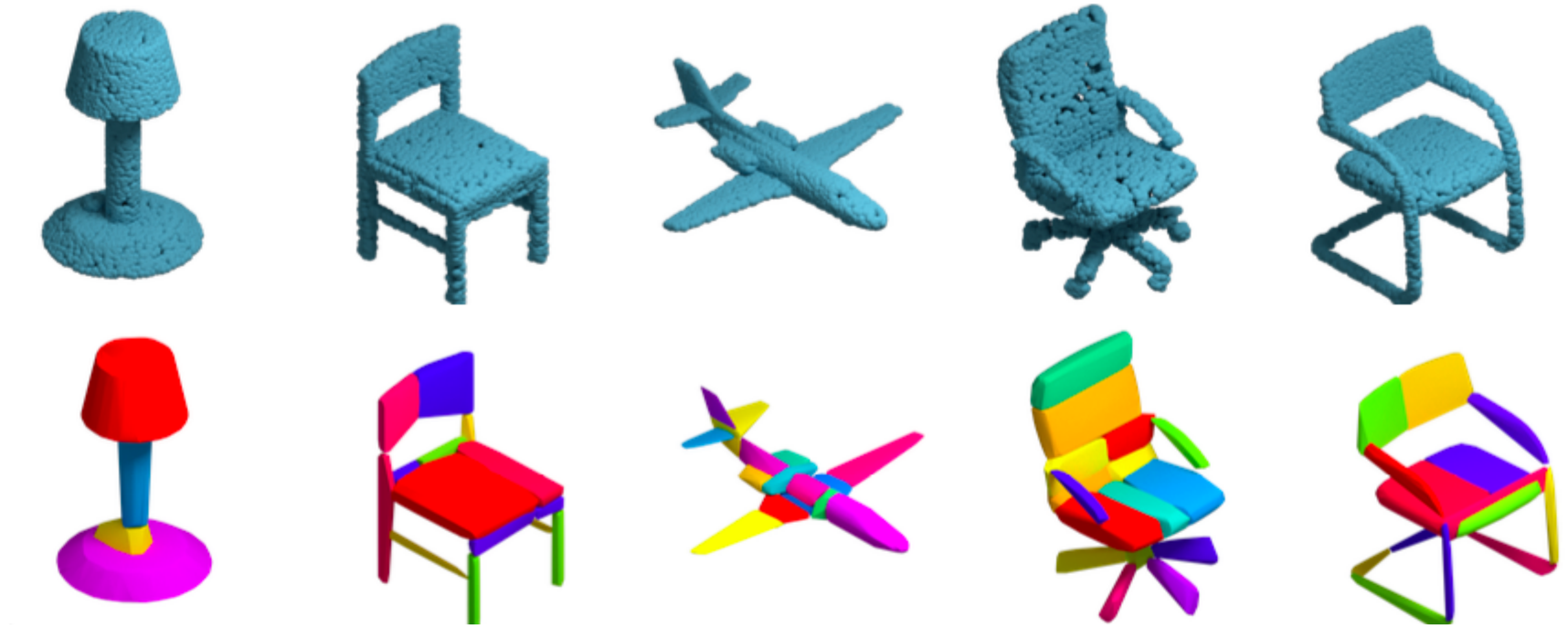
J. Walker et al. **An Uncertain Future: Forecasting from Static Images Using Variational Autoencoders**. ECCV 2016

3D shapes and convexity

- **Task:** Label 3D ***objects*** (chairs, tables..) into ***parts*** (legs, back, handles…)

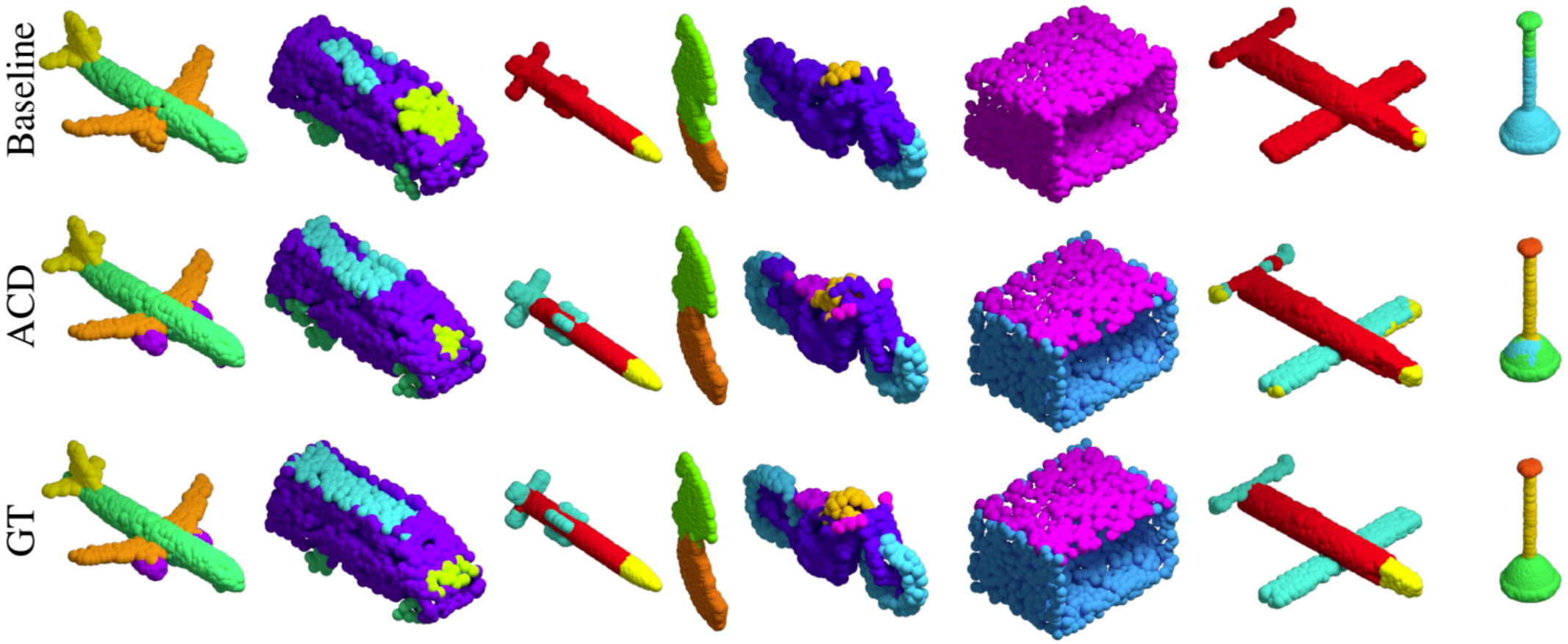
Approximate Convex Decomposition

- **Pretext Task:** Approximate Convex Decomposition
  - Get a large number of unlabeled 3D shapes
  - Run **off-the-shelf “ACD” software** to get decompositions
  - Train your favorite 3D neural network on this, and then apply on final task



[ECCV 2020]

10-Shot Segmentation Results



Gadelha and RoyChowdhury, et al., ECCV 2020

Large Language Models

pre-train transformers on text

Next-token-prediction

The model is given a sequence of words with the goal of predicting the next word.

Example:
Hannah is a ____

Hannah is a *sister*
Hannah is a *friend*
Hannah is a *marketer*
Hannah is a *comedian*

Masked-language-modeling

The model is given a sequence of words with the goal of predicting a 'masked' word in the middle.

Example
Jacob [mask] reading

Jacob *fears* reading
Jacob *loves* reading
Jacob *enjoys* reading
Jacob *hates* reading

Instruction tuning
Preference alignment

Finetuning

ChatGPT

https://twitter.com/thealexbanks/status/1624400398114234370

Summary of self-supervision via pretext-tasks

Pretext Tasks:

- Pretext tasks focus on “visual common sense”, e.g., rearrangement, predicting rotations, inpainting, colorization, etc.
- The models are forced learn good features about natural images, e.g., semantic representation of an object category, in order to solve the pretext tasks
- We don’t care about pretext task performance, but rather about the utility of the learned features for downstream tasks (classification, detection, segmentation)

Problems:

- Designing good pretext tasks is tedious and some kind of “art”
- The learned representations may not be general

Slides from Andreas Geiger, MPI Tubingen