



Computer Vision; Image Classification; Data-efficient Learning

Maziar Raissi

Assistant Professor

Department of Applied Mathematics

University of Colorado Boulder

maziar.raissi@colorado.edu



Boulder

Self-training with Noisy Student improves ImageNet classification

Require: Labeled images $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ and unlabeled images $\{\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_m\}$.

- 1: Learn teacher model θ_*^t which minimizes the cross entropy loss on labeled images

$$\frac{1}{n} \sum_{i=1}^n \ell(y_i, f^{noised}(x_i, \theta^t))$$

- 2: Use an unnoised teacher model to generate soft or hard pseudo labels for unlabeled images

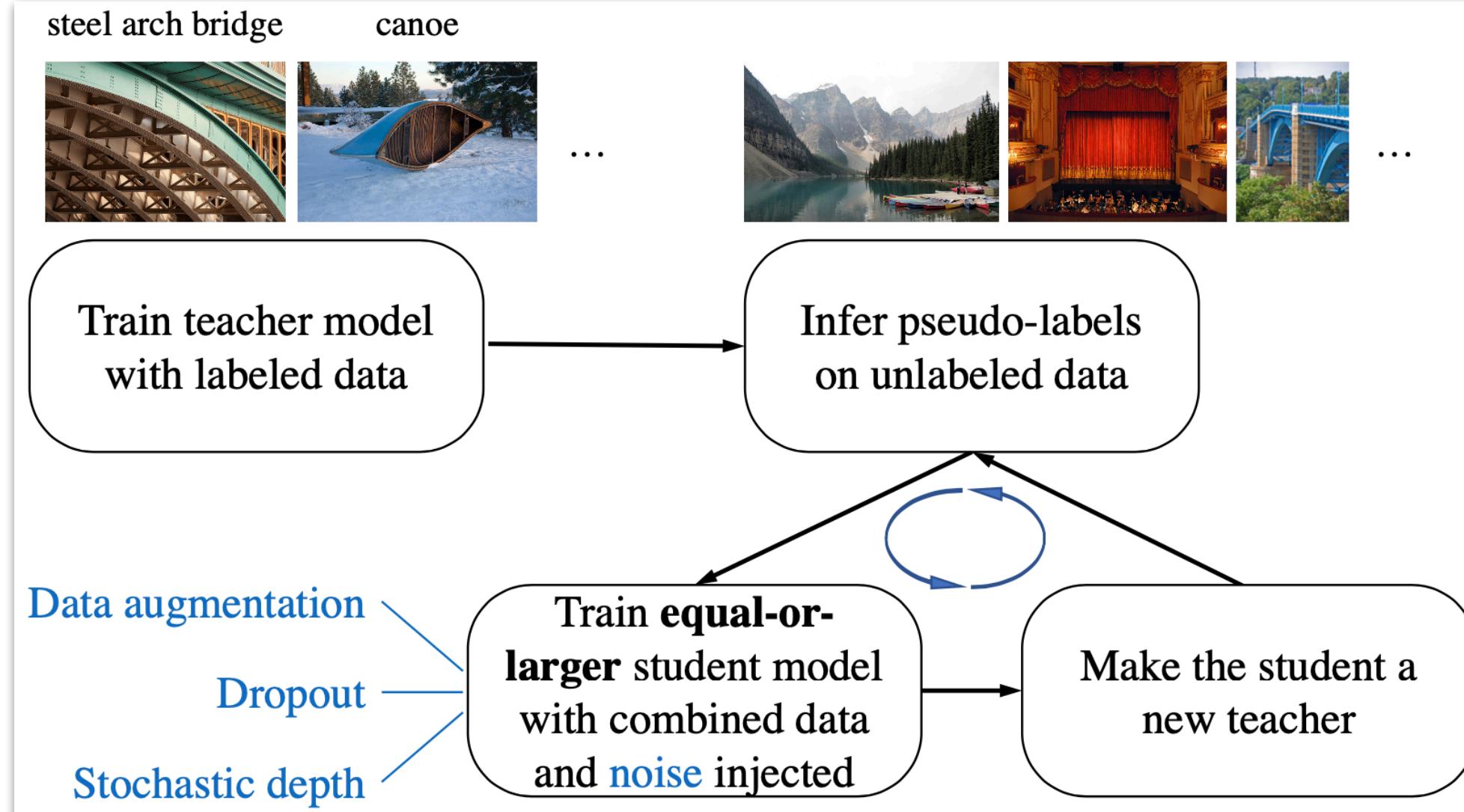
$$\tilde{y}_i = f(\tilde{x}_i, \theta_*^t), \forall i = 1, \dots, m$$

- 3: Learn an **equal-or-larger** student model θ_*^s which minimizes the cross entropy loss on labeled images and unlabeled images with **noise** added to the student model

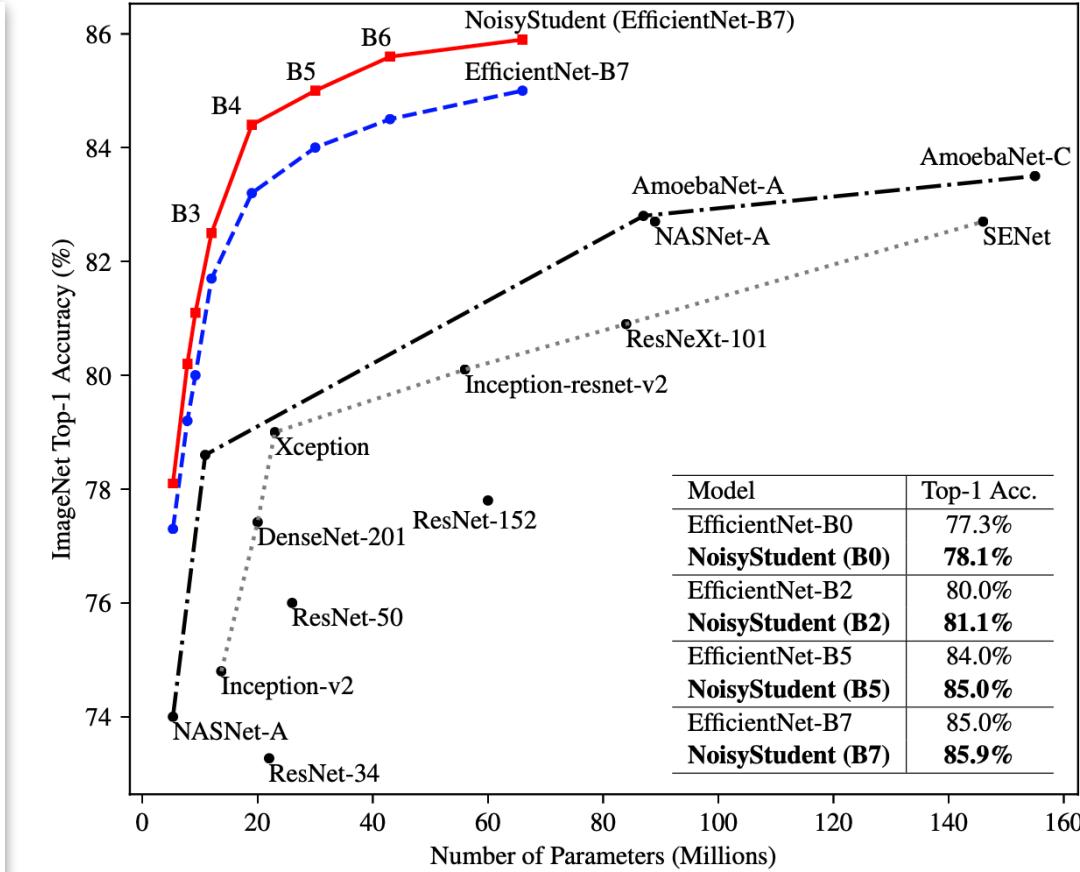
$$\frac{1}{n} \sum_{i=1}^n \ell(y_i, f^{noised}(x_i, \theta^s)) + \frac{1}{m} \sum_{i=1}^m \ell(\tilde{y}_i, f^{noised}(\tilde{x}_i, \theta^s))$$

- 4: Iterative training: Use the student as a teacher and go back to step 2.

	ImageNet top-1 acc.	ImageNet-A top-1 acc.	ImageNet-C mCE	ImageNet-P mFR
Prev. SOTA	86.4%	61.0%	45.7	27.8
NoisyStudent	88.4%	83.7%	28.3	12.2

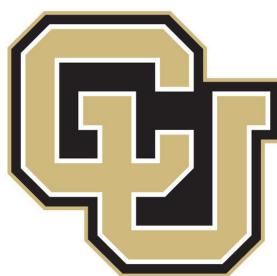


ImageNet-C and ImageNet-P test sets include images with common corruptions and perturbations such as blurring, fogging, rotation and scaling. ImageNet-A test set consists of difficult images that cause significant drops in accuracy to state-of-the-art models. These test sets are considered as “robustness” benchmarks.



mCE (mean corruption error) is the weighted average of error rate on different corruptions, with AlexNet’s error rate as a baseline (lower is better). mFR (mean flip rate) measures the model’s probability of flipping predictions under perturbations with AlexNet as a baseline (lower is better).





Boulder

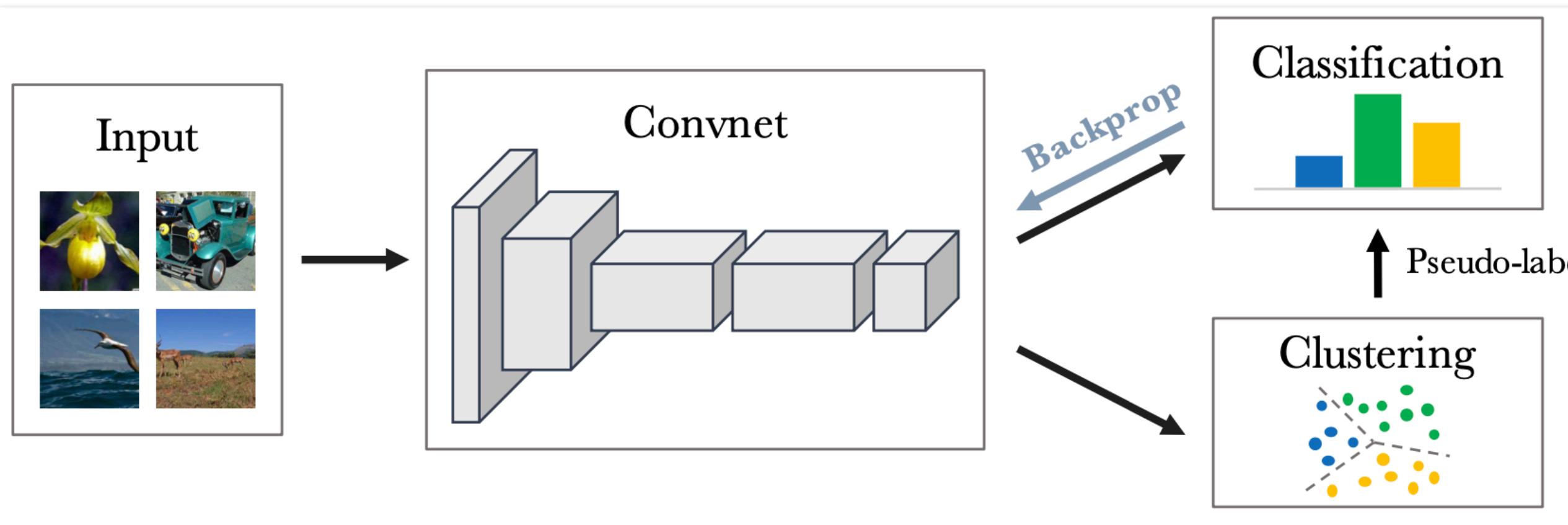
Deep Clustering for Unsupervised Learning of Visual Features

Key observation: A multilayer perceptron classifier on top of the last convolutional layer of a random AlexNet achieves 12% in accuracy on ImageNet while the chance is at 0.1%.

This is because a convolutional structure gives a strong prior.

$\{x_1, \dots, x_N\} \rightarrow$ training set of N images

Problem setup: Find parameters θ^* such that the convnet mapping f_{θ^*} produces good general-purpose features (representations).



k -means: cluster the features $f_{\theta}(x_n)$ into k distinct groups

$C \in \mathbb{R}^{d \times k} \rightarrow$ centroid matrix

$y_n \rightarrow$ cluster assignment of each image

$$\min_{C \in \mathbb{R}^{d \times k}} \frac{1}{N} \sum_{n=1}^N \min_{y_n \in \{0,1\}^k} \|f_{\theta}(x_n) - Cy_n\|_2^2 \quad \text{such that} \quad y_n^\top 1_k = 1.$$

$y_n^* \rightarrow$ optimal assignment (pseudo label)

$C^* \rightarrow$ optimal centroid matrix (not used)

$g_W \rightarrow$ parametrized classifier on top of the features $f_{\theta}(x_n)$

$$\min_{\theta, W} \frac{1}{N} \sum_{n=1}^N \ell(g_W(f_{\theta}(x_n)), y_n)$$

multinomial logistic loss

Avoiding Trivial Solutions

Empty clusters:

An optimal decision boundary is to assign all of the inputs to a single cluster.

When a cluster becomes empty, randomly select a non-empty cluster and use its centroid with a small random perturbation as the new centroid for the empty cluster. Then reassign the points belonging to the non-empty cluster to the two resulting clusters.

Trivial parametrization:

Unbalanced number of images per cluster

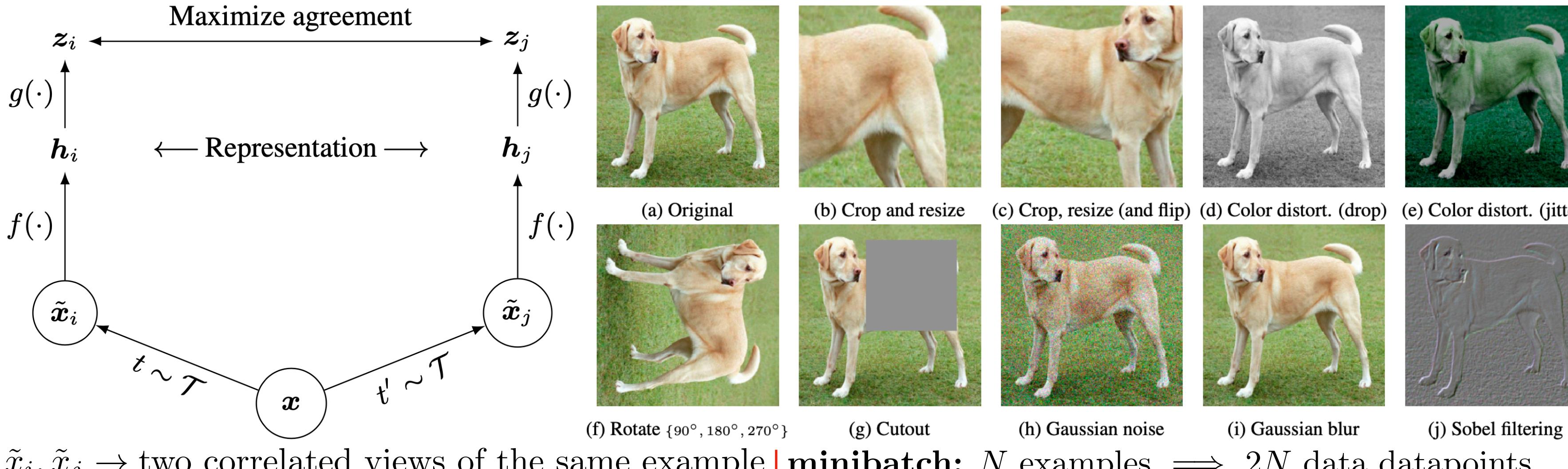
Sample images based on a uniform distribution over the classes, or pseudo-labels.

PASCAL VOC transfer tasks		Classification		Detection		Segmentation	
Method	Training set	FC6-8	ALL	FC6-8	ALL	FC6-8	ALL
Best competitor	ImageNet	63.0	67.7	43.4 [†]	53.2	35.8 [†]	37.7
DeepCluster	ImageNet	72.0	73.7	51.4	55.4	43.2	45.1
DeepCluster	YFCC100M	67.3	69.3	45.6	53.0	39.2	42.2

Pascal VOC 2007 object detection	Method	AlexNet		VGG-16		Method	Oxford5K		Paris6K	
		AlexNet	VGG-16	AlexNet	VGG-16		Oxford5K	Paris6K	Oxford5K	Paris6K
	ImageNet labels	56.8	67.3				72.4	81.5		
	Random	47.8	39.7				6.9	22.0		
	Doersch <i>et al.</i> [13]	51.1	61.5				35.4	53.1		
	Wang and Gupta [63]	47.2	60.2				42.3	58.0		
	Wang <i>et al.</i> [64]	–	63.2							
	DeepCluster	55.4	65.9				61.0	72.0		

mAP on instance-level image retrieval on Oxford and Paris dataset with a VGG-16

A Simple Framework for Contrastive Learning of Visual Representations



Data Augmentation

Random cropping and resizing to the original size

Random color distortions

Random Gaussian blur

Base Encoder

$$h_i = f(\tilde{x}_i) = \text{ResNet}(\tilde{x}_i)$$

$h_i \in \mathbb{R}^d \rightarrow$ output after average pooling layer

Projection Head

$$z_i = g(h_i) = W^2 \sigma(W^1 h_i)$$

ReLU

Contrastive Loss

Contrastive Prediction Task: Given a set $\{\tilde{x}_k\}$ including a positive pair of examples \tilde{x}_i and \tilde{x}_j , identify \tilde{x}_j in $\{\tilde{x}_k\}_{k \neq i}$ for a given \tilde{x}_i .

minibatch: N examples $\implies 2N$ data datapoints

Given a positive pair, treat the other $2(N - 1)$ augmented examples as negatives

$$\text{sim}(u, v) = u^T v / \|u\| \|v\| \rightarrow \text{cosine similarity}$$

$$\ell_{i,j} = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(z_i, z_k)/\tau)}$$

NT-Xent Loss (normalized temperature-scaled cross entropy loss)

$\tau \rightarrow$ temperature

The final loss is computed over all positive pairs (i, j) & (j, i) .

Transfer learning performance

	Food	CIFAR10	CIFAR100	Birdsnap	SUN397	Cars	Aircraft	VOC2007	DTD	Pets	Caltech-101	Flowers
<i>Linear evaluation:</i>												
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
<i>Fine-tuned:</i>												
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

Self-supervised learning on ImageNet

Method	Architecture	Param (M)	Top 1	Top 5
<i>Methods using ResNet-50:</i>				
Local Agg.	ResNet-50	24	60.2	-
MoCo	ResNet-50	24	60.6	-
PIRL	ResNet-50	24	63.6	-
CPC v2	ResNet-50	24	63.8	85.3
SimCLR (ours)	ResNet-50	24	69.3	89.0

Methods using other architectures:

Rotation	RevNet-50 (4×)	86	55.4	-
BigBiGAN	RevNet-50 (4×)	86	61.3	81.9
AMDIM	Custom-ResNet	626	68.1	-
CMC	ResNet-50 (2×)	188	68.4	88.2
MoCo	ResNet-50 (4×)	375	68.6	-
CPC v2	ResNet-161 (*)	305	71.5	90.1
SimCLR (ours)	ResNet-50 (2×)	94	74.2	92.0
SimCLR (ours)	ResNet-50 (4×)	375	76.5	93.2

Semi-supervised learning on ImageNet

Method	Architecture	Label fraction	
		1%	10%
Supervised baseline	ResNet-50	48.4	80.4
Top 5			

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

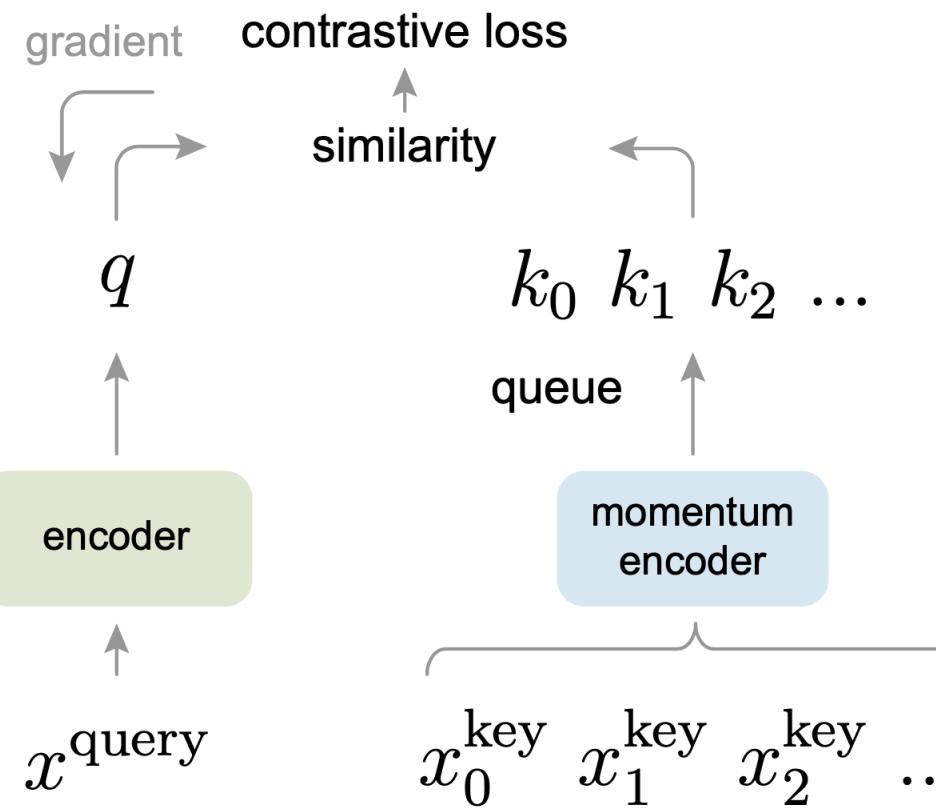
Methods using representation learning only:

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



Boulder

Momentum Contrast for Unsupervised Visual Representation Learning



Keys (tokens): samples from the data (e.g., images or patches) represented by an encoder network

Unsupervised Learning:

dictionary look-up

An encoded "query" should be similar to its matching "key" and dissimilar to others

Instance discrimination task

A query matches a key if they are encoded views (e.g., different crops) of the same image

Contrastive learning

$q \rightarrow$ encoded query

$\{k_0, k_1, \dots\} \rightarrow$ set of encoded samples (keys of a dictionary)

$$k_+ \rightarrow \text{the single key in the dictionary that matches } q$$

$$\mathcal{L}_q = -\log \frac{\exp(q \cdot k_+ / \tau)}{\sum_{i=0}^K \exp(q \cdot k_i / \tau)}$$

InfoNCE (noise-contrastive estimation) loss

$\tau \rightarrow$ temperature hyper-parameter

$q = f_q(x^q)$ query sample

$k = f_k(x^k)$ encoder network

Momentum Contrast (MoCo)

- ① dynamic
 - ② large
 - ③ consistent dictionary
- dictionary as queue

The current mini-batch is enqueue to the dictionary, and the oldest mini-batch in the queue is removed

$\theta_k \rightarrow$ parameters of f_k

$\theta_q \rightarrow$ parameters of f_q

$\theta_k \leftarrow m\theta_k + (1-m)\theta_q, m \in [0, 1], m = 0.999$

$\theta_q \rightarrow$ updated by backprop

Object detection	fine-tuned on PASCAL VOC		
pre-train	AP ₅₀	AP	AP ₇₅
random init.	60.2	33.8	33.1
super. IN-1M	81.3	53.5	58.8
MoCo IN-1M	81.5 (+0.2)	55.9 (+2.4)	62.6 (+3.8)
MoCo IG-1B	82.2 (+0.9)	57.2 (+3.7)	63.7 (+4.9)

Faster R-CNN

ImageNet-1M (IN-1M), Instagram-1B (IG-1B), super. (supervised)

Algorithm 1 Pseudocode of MoCo in a PyTorch-like style.

```

# f_q, f_k: encoder networks for query and key
# queue: dictionary as a queue of K keys (CxK)
# m: momentum
# t: temperature

f_k.params = f_q.params # initialize
for x in loader: # load a minibatch x with N samples
    x_q = aug(x) # a randomly augmented version
    x_k = aug(x) # another randomly augmented version

    q = f_q.forward(x_q) # queries: Nx1
    k = f_k.forward(x_k) # keys: Nx1
    k = k.detach() # no gradient to keys

    # positive logits: Nx1
    l_pos = bmm(q.view(N,1,C), k.view(N,C,1))

    # negative logits: NxK
    l_neg = mm(q.view(N,C), queue.view(C,K))

    # logits: Nx(1+K)
    logits = cat([l_pos, l_neg], dim=1)

    # contrastive loss, Eqn. (1)
    labels = zeros(N) # positives are the 0-th
    loss = CrossEntropyLoss(logits/t, labels)

    # SGD update: query network
    loss.backward()
    update(f_q.params)

    # momentum update: key network
    f_k.params = m*f_k.params+(1-m)*f_q.params

    # update dictionary
    enqueue(queue, k) # enqueue the current minibatch
    dequeue(queue) # dequeue the earliest minibatch

```

bmm: batch matrix multiplication; mm: matrix multiplication; cat: concatenation.



Boulder

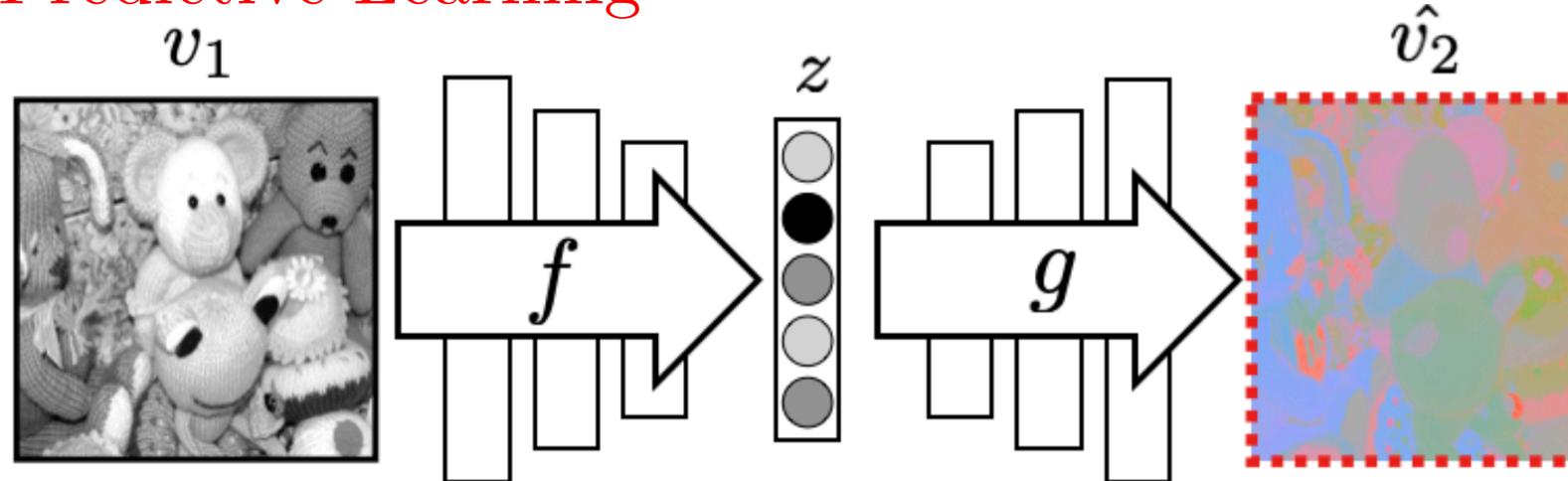
Contrastive Multiview Coding

A “dog” can be seen, heard and felt!

$V_1, V_2, \dots, V_M \rightarrow$ a collection of M views of the data

$v_i \sim \mathcal{P}(V_i) \rightarrow$ a random variable representing samples

Predictive Learning



$V_1, V_2 \rightarrow$ two views of a dataset
(e.g., luminance and chrominance)

$z = f(v_1) \rightarrow$ encoder

$\hat{v}_2 = g(z) \rightarrow$ decoder

↳ prediction of v_2 given v_1

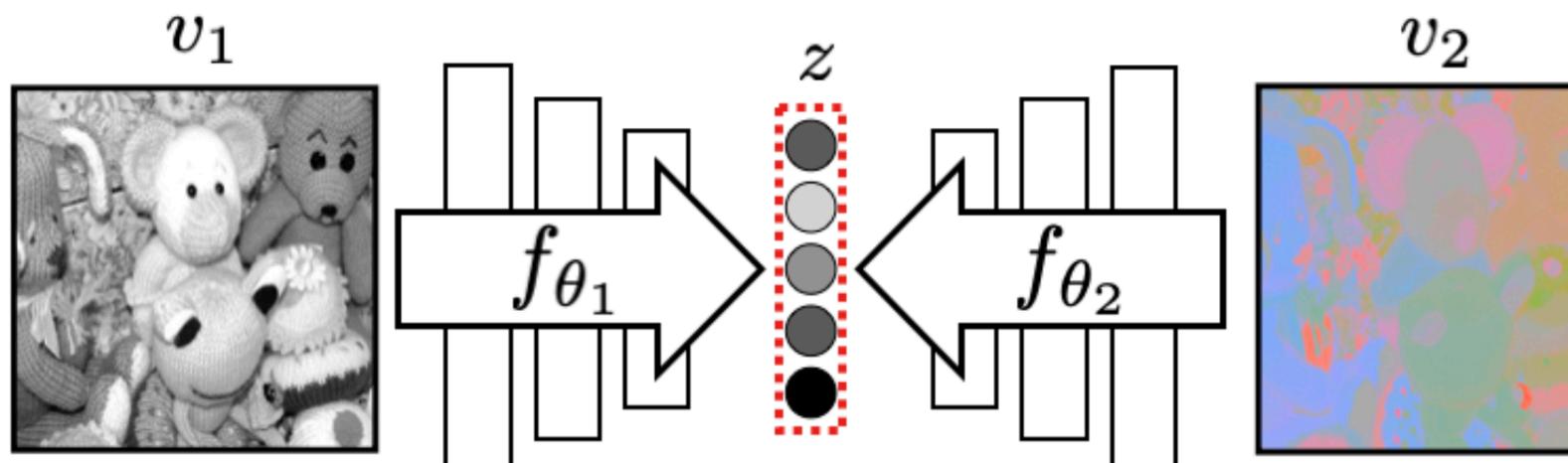
$z \rightarrow$ latent variables

Bring \hat{v}_2 close to $v_2 \rightarrow$ objective function (L_1 or L_2)

Independence assumption between elements or pixels

$p(v_2|v_1) = \prod_i p(v_{2i}|v_1)$ of v_2 given v_1 !

Contrastive Learning with Two Views



$\{v_1^i, v_2^i\}_{i=1}^N \rightarrow$ collection of samples from V_1 and V_2

$x = \{v_1^i, v_2^i\} \rightarrow$ positives (congruent pairs), $x \sim p(v_1, v_2) \rightarrow$ joint distribution

$y = \{v_1^i, v_2^j\} \rightarrow$ negatives (incongruent pairs), $y \sim p(v_1)p(v_2) \rightarrow$ product of marginals

Select a single positive sample x out of a set $S = \{x, y_1, y_2, \dots, y_k\}$ that contains k negative samples

$$\mathcal{L}_{contrast} = -\mathbb{E}_S \left[\log \frac{h_\theta(x)}{h_\theta(x) + \sum_{i=1}^k h_\theta(y_i)} \right]$$

$$h_\theta(\{v_1, v_2\}) = \exp\left(\frac{f_{\theta_1}(v_1) \cdot f_{\theta_2}(v_2)}{\|f_{\theta_1}(v_1)\| \cdot \|f_{\theta_2}(v_2)\|} \cdot \frac{1}{\tau}\right)$$

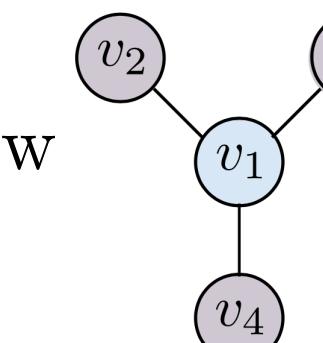
Connecting to Mutual Information

$h_\theta^*(\{v_1, v_2\}) \propto \frac{p(z_1, z_2)}{p(z_1)p(z_2)} \propto \frac{p(z_1|z_2)}{p(z_1)} \rightarrow$ point-wise mutual information

$$I(v_i; v_j) \geq I(z_i; z_j) \geq \log(k) - \mathcal{L}_{contrast}$$

Contrastive Learning with More Than Two Views

$$\mathcal{L}_C = \sum_{j=2}^M \mathcal{L}(V_1, V_j) \rightarrow \text{core view}$$



$$\mathcal{L}_F = \sum_{1 \leq i < j \leq M} \mathcal{L}(V_i, V_j) \rightarrow \text{full graph}$$

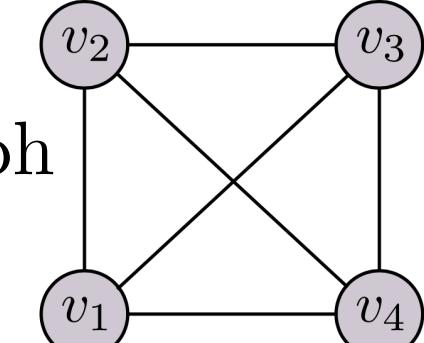


Image Views

luminance (L channel), chrominance (ab channel)

Video Views

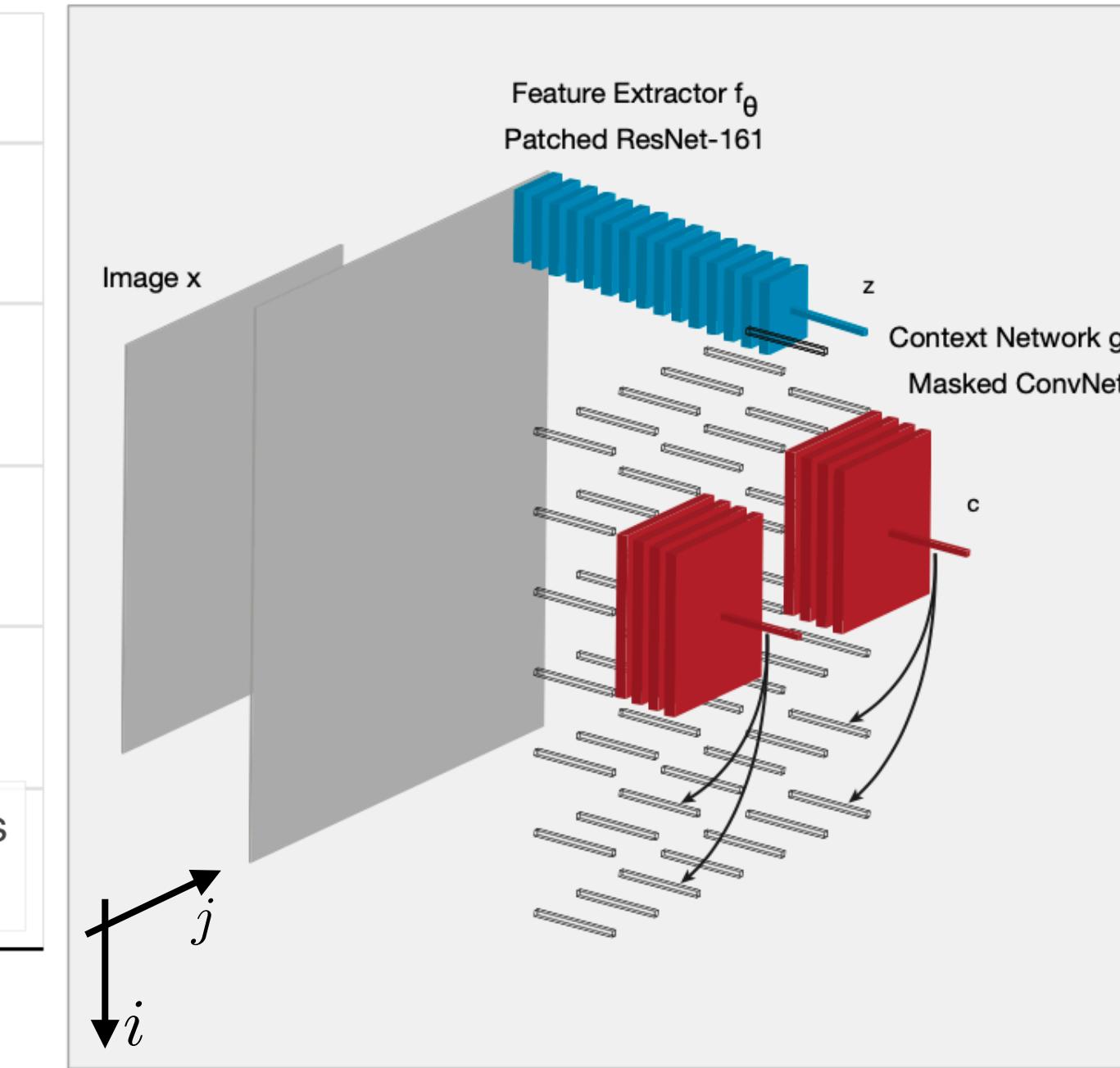
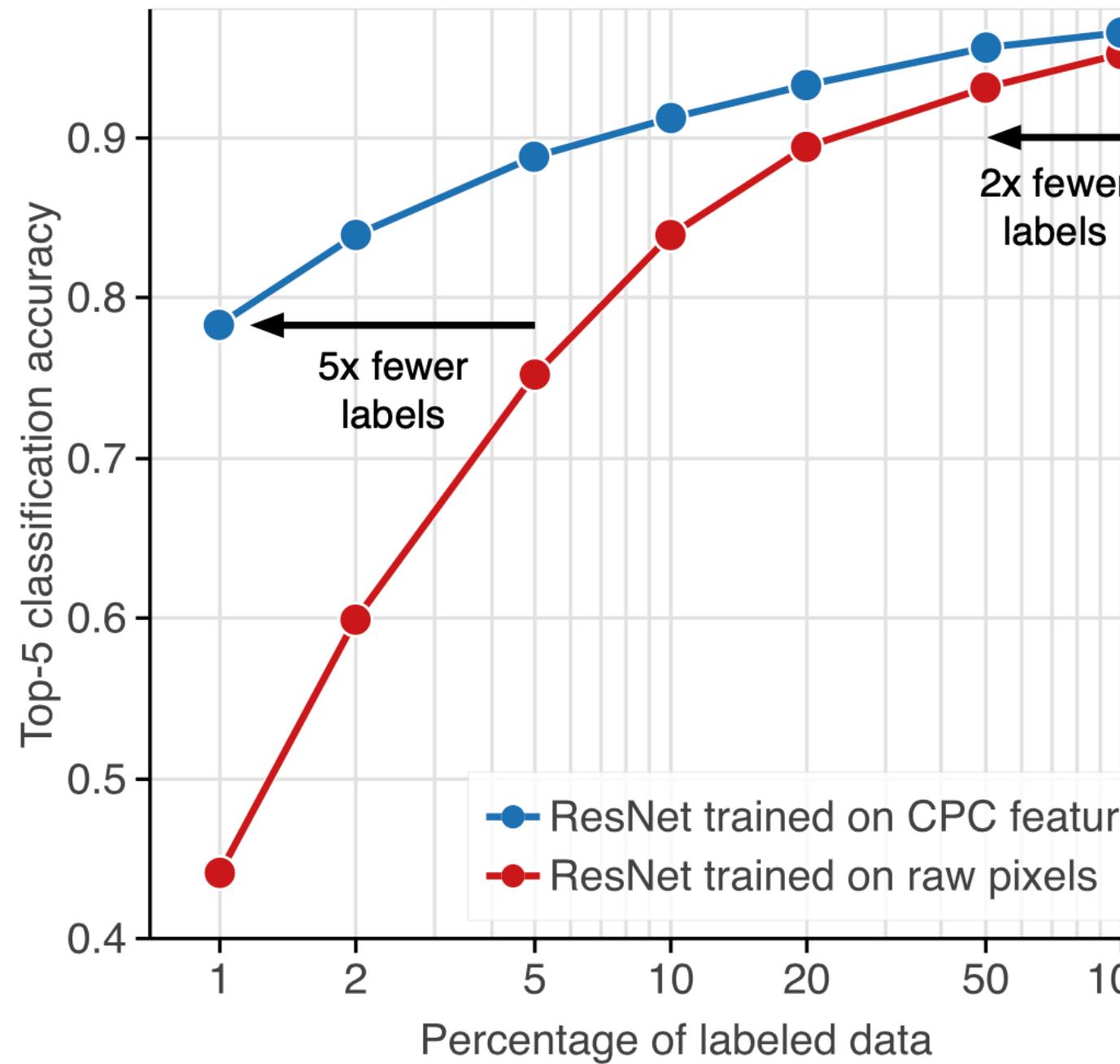
$i_t \rightarrow$ image at time t , $i_{t+k} \rightarrow$ neighboring frame, $f_t \rightarrow$ optical flow (10 frames centered at time t)
 $(i_t, i_{t+k}), (i_t, f_t) \rightarrow$ two separate contrastive learning objectives

Multiple Image Views

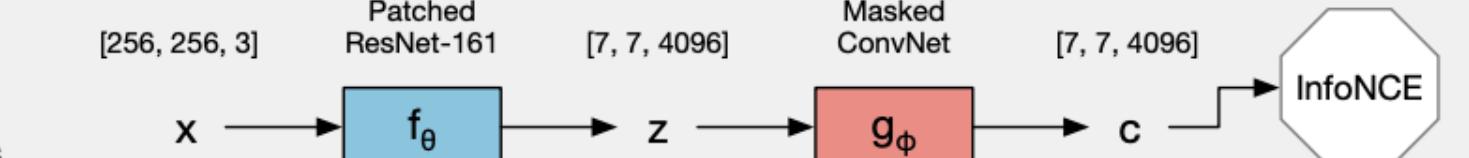
luminance (L channel), chrominance (ab channel), depth, surface normal, and semantic labels



Data-Efficient Image Recognition with Contrastive Predictive Coding



Self-supervised pre-training
100% images; 0% labels



Linear classification
100% images and labels



Efficient classification
1% to 100% images and labels



Transfer learning
100% images and labels



Supervised training
1% to 100% images and labels



$\{x_{i,j}\} \rightarrow$ a grid of overlapping patches dividing each input image
 $i, j \rightarrow$ location of the patch

$z_{i,j} = f_\theta(x_{i,j}) \rightarrow$ encoded patch
 ↗ neural network

$g_\phi \rightarrow$ masked convolutional network (applied to the grid of feature vectors)

$c_{i,j} = g_\phi(\{z_{u,v}\}_{u \leq i, v}) \rightarrow$ context vector
 ↗ feature vectors that lie above i, j

$z_{i+k,j} \rightarrow$ “future” feature vectors to be predicted from current context vector $c_{i,j}$
 $k \rightarrow$ prediction length

$\hat{z}_{i+k,j} = W_k c_{i,j}$
 ↗ prediction matrix

– increase depth & width – layer normalization – making predictions in all four directions – extensive patch-based data augmentation

Henaff, Olivier. "Data-efficient image recognition with contrastive predictive coding." *International Conference on Machine Learning*. PMLR, 2020.

Correctly recognize the target $z_{i+k,j}$ among a set of randomly sampled feature vectors $\{z_l\}$ from the dataset (i.e., other locations in the image and other images in the mini-batch)

$$\begin{aligned} \mathcal{L}_{\text{CPC}} &= - \sum_{i,j,k} \log p(z_{i+k,j} | \hat{z}_{i+k,j}, \{z_l\}) \\ &= - \sum_{i,j,k} \log \frac{\exp(\hat{z}_{i+k,j}^T z_{i+k,j})}{\exp(\hat{z}_{i+k,j}^T z_{i+k,j}) + \sum_l \exp(\hat{z}_{i+k,j}^T z_l)} \end{aligned}$$

InfoNCE (Noise Contrastive Estimation)



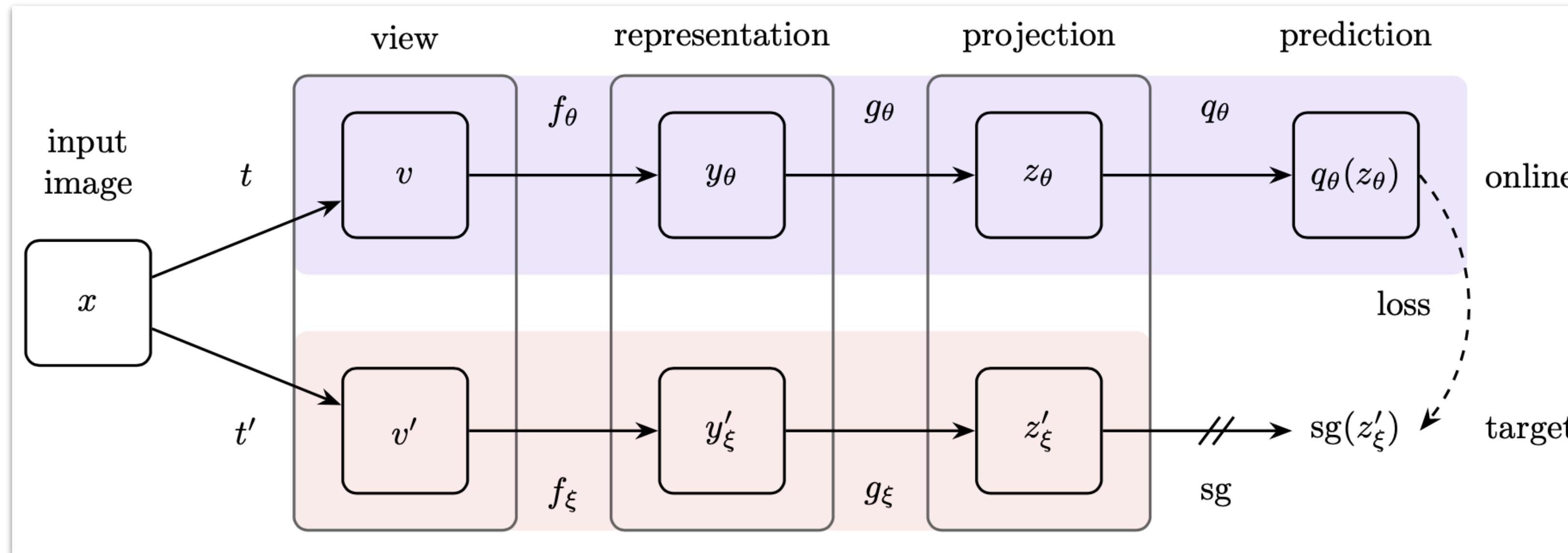
Boulder

Bootstrap your own latent: A new approach to self-supervised Learning

Bootstrap Your Own Latent (BYOL)

Self-supervised image representation learning without using negative pairs! (★)

Learn a representation y_θ which can be used for downstream tasks!



Three stages of the online network:

1. $f_\theta \rightarrow$ encoder
2. $g_\theta \rightarrow$ projector
3. $q_\theta \rightarrow$ predictor (★)

$\theta \rightarrow$ online parameters

$\xi \rightarrow$ parameters of the target network

$\xi \leftarrow \tau\xi + (1 - \tau)\theta$ (★)

$\mathcal{D} \rightarrow$ set of images

$x \sim \mathcal{D} \rightarrow$ an image sampled uniformly from \mathcal{D}

$\mathcal{T}, \mathcal{T}' \rightarrow$ two distributions of image augmentations

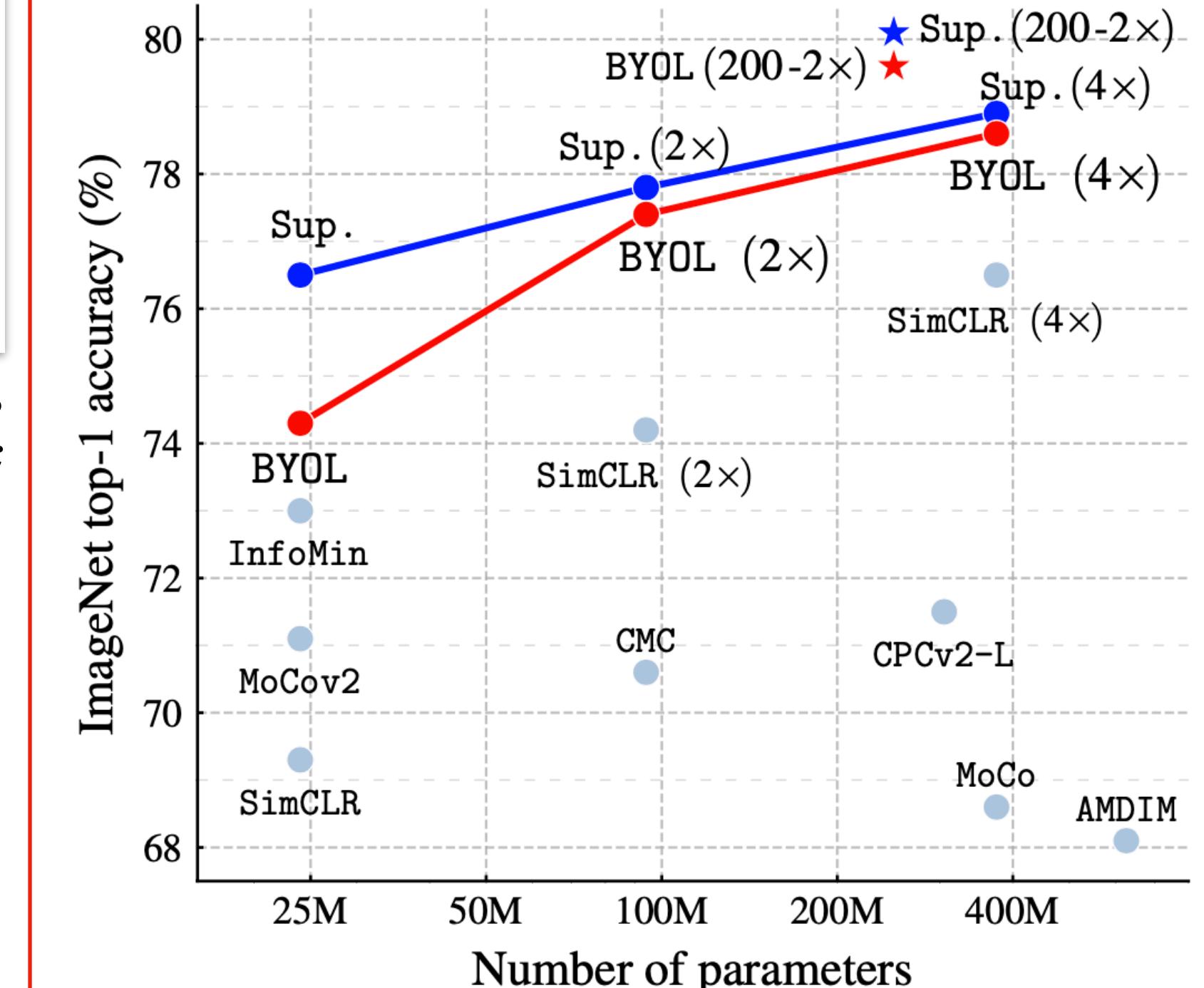
$v = t(x), v' = t'(x) \rightarrow$ two augmented views from x
 $t \sim \mathcal{T}, t' \sim \mathcal{T}'$
 $y_\theta = f_\theta(v) \rightarrow$ representation
 $z_\theta = g_\theta(y_\theta) \rightarrow$ projection
 $y'_\xi = f'_\xi(v') \rightarrow$ target representation
 $z'_\xi = g'_\xi(y'_\xi) \rightarrow$ target projection
 $q_\theta(z_\theta) \rightarrow$ prediction of z'_ξ
 $\bar{q}_\theta(z_\theta), \bar{z}'_\xi \rightarrow l_2$ normalized

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

$$\mathcal{L}_{\text{BYOL}} = \mathcal{L}_{\theta, \xi} + \tilde{\mathcal{L}}_{\theta, \xi}$$

feeding v' to the online network
and v to the target network

While this objective admits collapsed solutions, e.g., outputting the same vector for all images, the paper empirically shows that BYOL does not converge to such solutions. It hypothesizes (see Section 3.2) that the combination of (i) the addition of a predictor to the online network and (ii) the use of a slow-moving average of the online parameters as the target network encourages encoding more and more information within the online projection and avoids collapsed solutions.



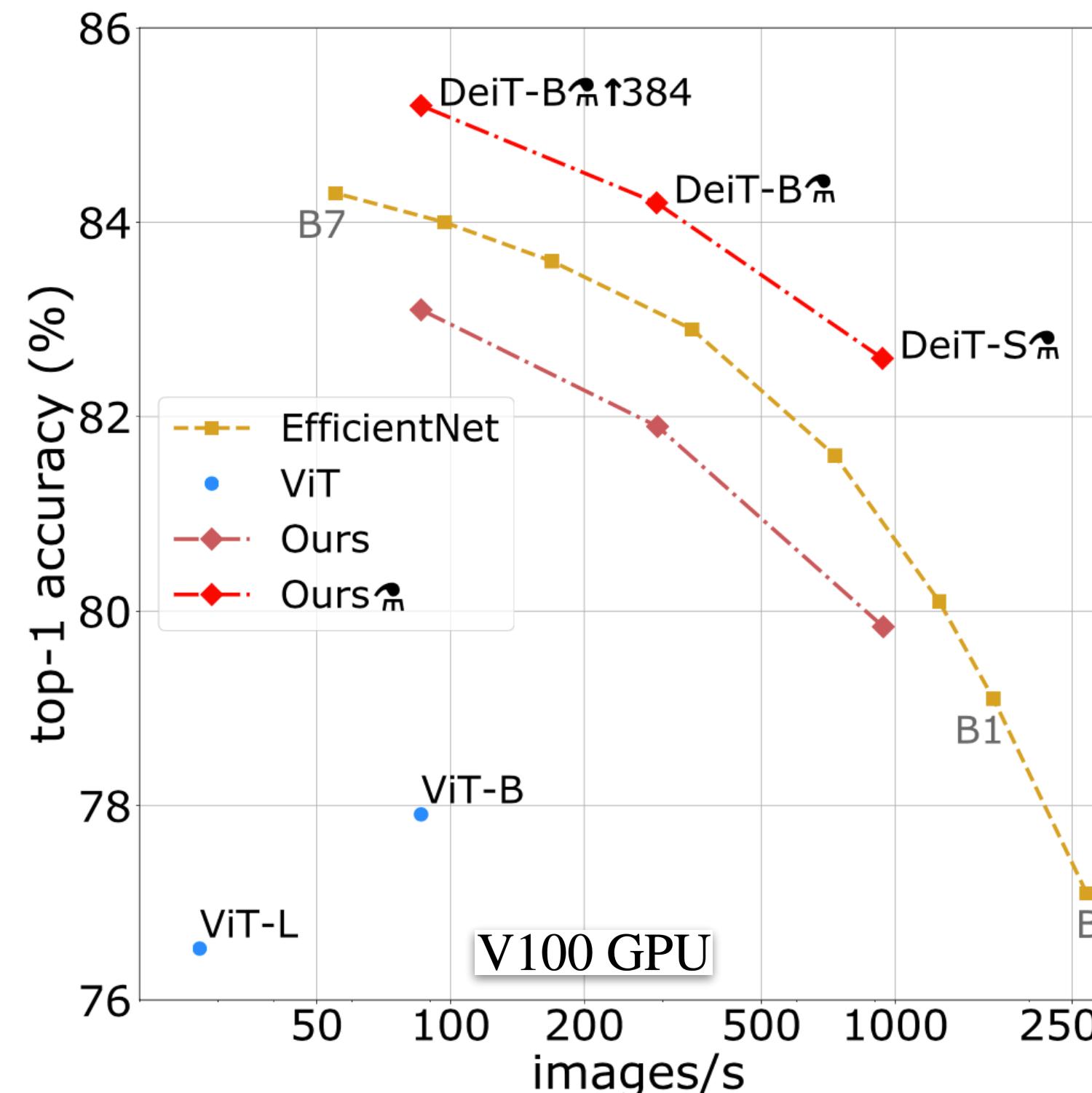


Boulder

Training Data-Efficient Image Transformers & Distillation Through Attention

DeiT: Data-Efficient Image Transformers

ImageNet Data Only (No External Data such as JFT-300M)



→ models trained with our transformer-specific distillation

Fixing the positional encoding across resolutions

Use a lower training resolution and fine-tune the network at a larger resolution

Keep the image patch sizes the same

⇒ N (sequence length) changes

⇒ need to adapt positional encodings (use bicubic interpolation)

Pre-training	Fine-tuning	Rand-Augment				Stoch. Depth	Repeated Aug.	Exp. Moving Avg.	pre-trained 224	fine-tuned 384
		AutoAug	Mixup	CutMix	Erasing					
adamw	adamw	✓	✗	✓	✓	✓	✓	✗	81.8 \pm 0.2	83.1 \pm 0.1
SGD	adamw	✓	✗	✓	✓	✓	✓	✗	74.5	77.3
adamw	SGD	✓	✗	✓	✓	✓	✓	✗	81.8	83.1
adamw	adamw	✗	✗	✓	✓	✓	✓	✗	79.6	80.4
adamw	adamw	✗	✓	✓	✓	✓	✓	✗	81.2	81.9
adamw	adamw	✓	✗	✗	✓	✓	✓	✗	78.7	79.8
adamw	adamw	✓	✗	✓	✗	✓	✓	✗	80.0	80.6
adamw	adamw	✓	✗	✗	✓	✓	✓	✗	75.8	76.7
adamw	adamw	✓	✗	✓	✓	✓	✓	✗	4.3*	0.1
adamw	adamw	✓	✗	✓	✓	✓	✗	✓	3.4*	0.1
adamw	adamw	✓	✗	✓	✓	✓	✗	✓	76.5	77.4
adamw	adamw	✓	✗	✓	✓	✓	✓	✓	81.3	83.1
adamw	adamw	✓	✗	✓	✓	✓	✓	✓	81.9	83.1

Distillation through attention

Teacher Model: A strong image classifier
(e.g., RegNetY ConvNet)

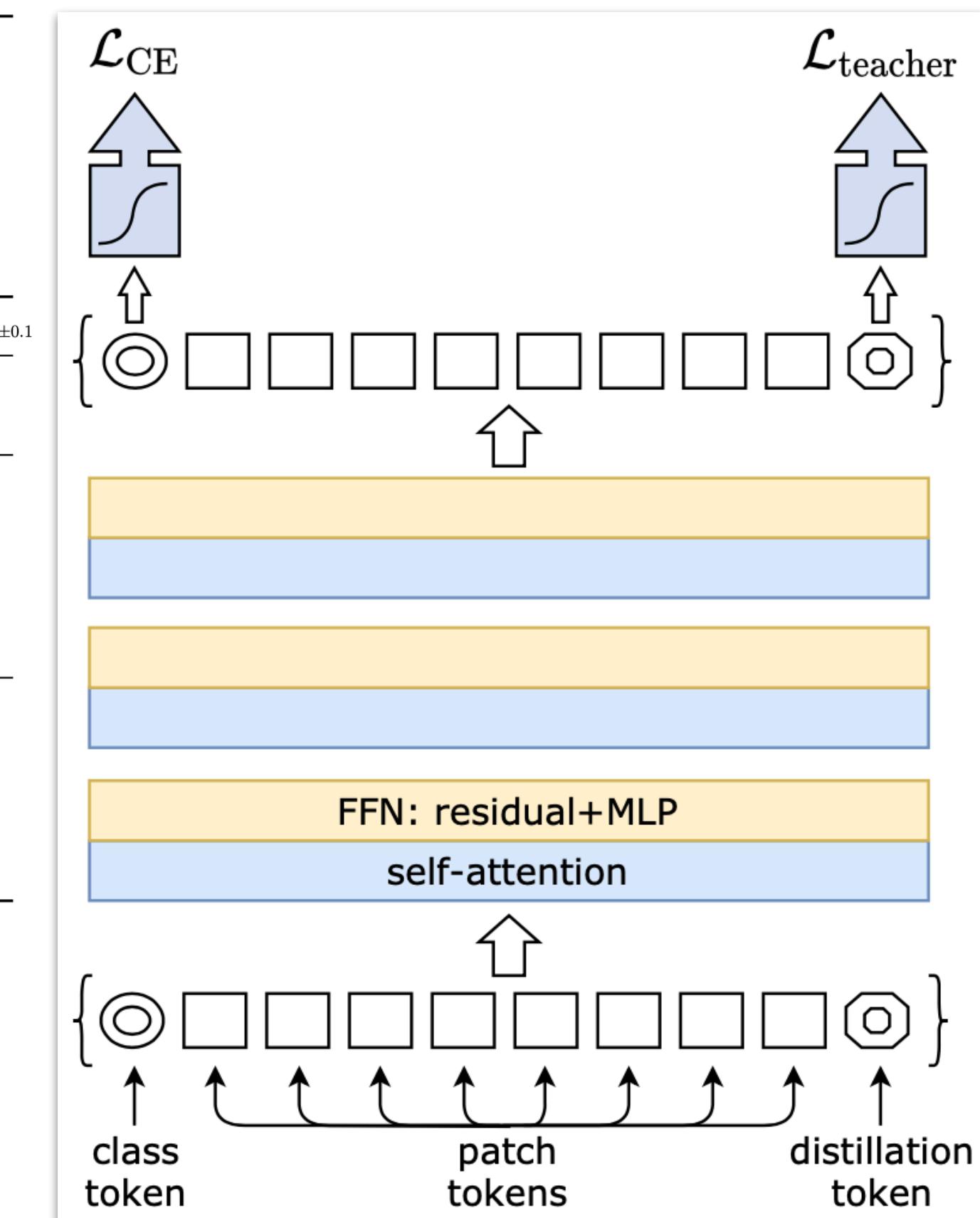
Soft Distillation

$$\begin{aligned} \mathcal{L}_{\text{global}} = & (1 - \lambda) \mathcal{L}_{\text{CE}}(\psi(Z_s), y) \\ & + \lambda \tau^2 \text{KL}(\psi(Z_s/\tau), \psi(Z_t/\tau)) \end{aligned}$$

$\psi \rightarrow \text{softmax}$

$\tau \rightarrow \text{temperature}$

$Z_s, Z_t \rightarrow \text{student and teacher logits}$



Hard-label Distillation

$$y_t = \text{argmax}_c Z_t(c)$$

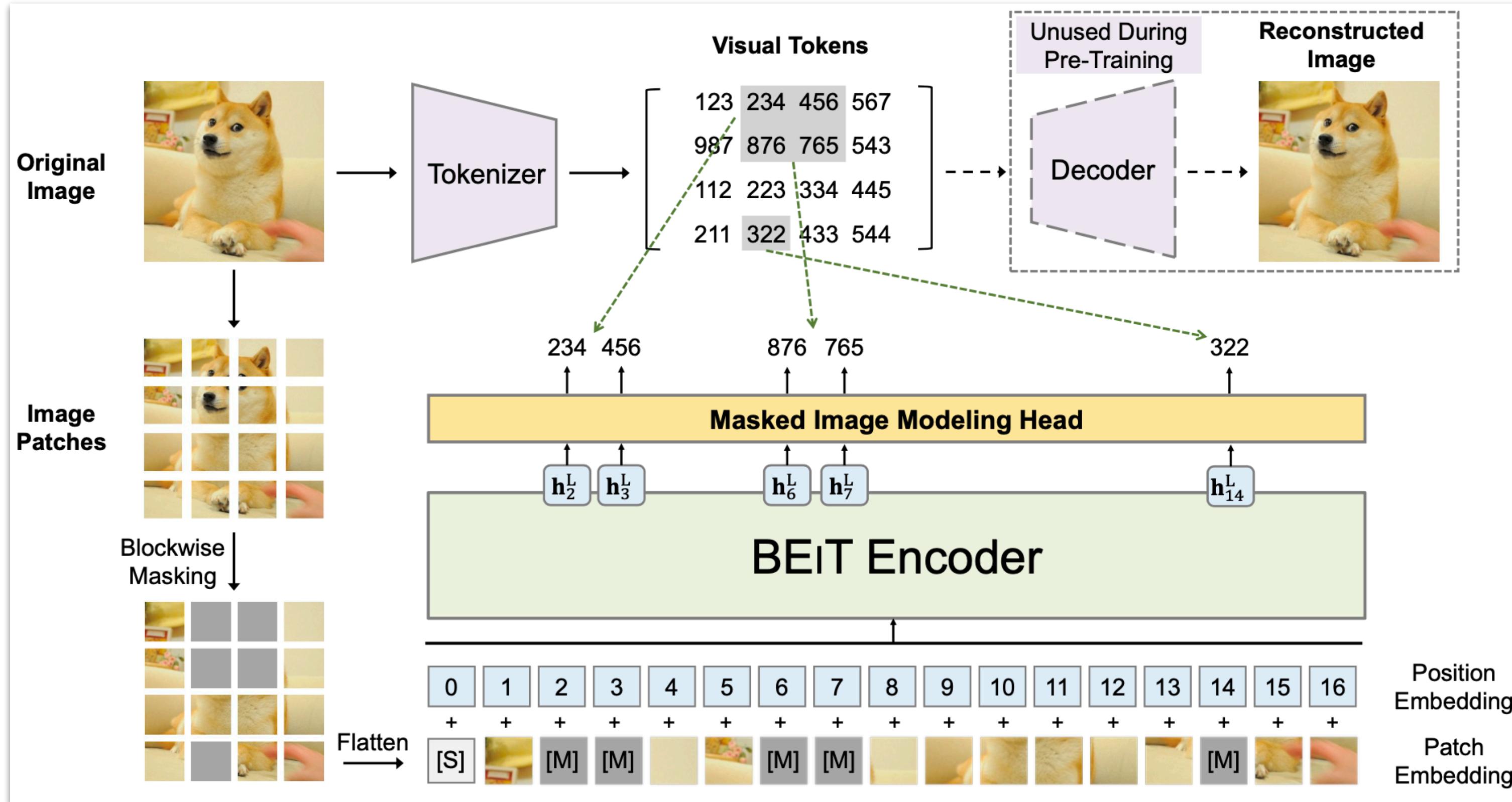
$$\begin{aligned} \mathcal{L}_{\text{global}}^{\text{hardDistill}} = & \frac{1}{2} \mathcal{L}_{\text{CE}}(\psi(Z_s), y) \\ & + \frac{1}{2} \mathcal{L}_{\text{CE}}(\psi(Z_s), y_t) \end{aligned}$$



Boulder

BEiT: BERT Pre-Training of Image Transformers

Bidirectional Encoder representation from Image Transformers



Visual Token

image tokenizer (discrete variational autoencoder)

tokenize $x \in \mathbb{R}^{H \times W \times C}$ into $z \in \mathcal{V}^{h \times w}$

$\mathcal{V} = \{1, \dots, |\mathcal{V}|\} \rightarrow$ vocabulary containing discrete token indices

<https://github.com/openai/DALL-E>

$q_\phi(z|x) \rightarrow$ tokenizer (maps image x into discrete tokens z according to a visual codebook/vocabulary)

Bao, Hangbo, Li Dong, and Furu Wei. "BEiT: BERT Pre-Training of Image Transformers." *arXiv preprint arXiv:2106.08254* (2021).

$p_\psi(x|z) \rightarrow$ decoder (reconstruct the input image x based on the visual tokens z)

$$\max_{\phi, \psi} \mathbb{E}_{z \sim q_\phi(z|x)} [\log p_\psi(x|z)] - \text{KL}[q_\phi(z|x) \parallel p(z)]$$

Gumbel-softmax relaxation

Algorithm 1 Blockwise Masking

Input: $N (= h \times w)$ image patches

Output: Masked positions \mathcal{M}

$\mathcal{M} \leftarrow \{\}$

repeat

$s \leftarrow \text{Rand}(16, 0.4N - |\mathcal{M}|)$

$r \leftarrow \text{Rand}(0.3, \frac{1}{0.3})$

$a \leftarrow \sqrt{s \cdot r}; b \leftarrow \sqrt{s/r}$

$t \leftarrow \text{Rand}(0, h - a); l \leftarrow \text{Rand}(0, w - b)$

$\mathcal{M} \leftarrow \mathcal{M} \cup \{(i, j) : i \in [t, t + a], j \in [l, l + b]\}$

until $|\mathcal{M}| > 0.4N$

\triangleright Masking ratio is 40%

return \mathcal{M}

Models	CIFAR-100	ImageNet
<i>Training from scratch (i.e., random initialization)</i>		
ViT ₃₈₄ (Dosovitskiy et al., 2020)	48.5*	77.9
DeiT (Touvron et al., 2020)	n/a	81.8
<i>Supervised Pre-Training on ImageNet-1K (using labeled data)</i>		
ViT ₃₈₄ (Dosovitskiy et al., 2020)	87.1	77.9
DeiT (Touvron et al., 2020)	90.8	81.8
<i>Self-Supervised Pre-Training on ImageNet-1K (without labeled data)</i>		
iGPT-1.36B [†] (Chen et al., 2020a)	n/a	66.5
ViT ₃₈₄ -JFT300M [‡] (Dosovitskiy et al., 2020)	n/a	79.9
DINO (Caron et al., 2021)	91.7	82.8
MoCo v3 (Chen et al., 2021)	87.1	n/a
BEiT (ours)	90.1	83.2
<i>Self-Supervised Pre-Training, and Intermediate Fine-Tuning on ImageNet-1K</i>		
BEiT (ours)	91.8	83.2

Top-1 accuracy of image classification



Boulder

Questions?
