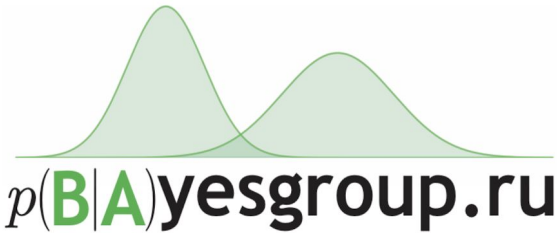
Contrastive Self-Supervised Learning for Visual Representations (CPC and SimCLR)

Andrei Atanov

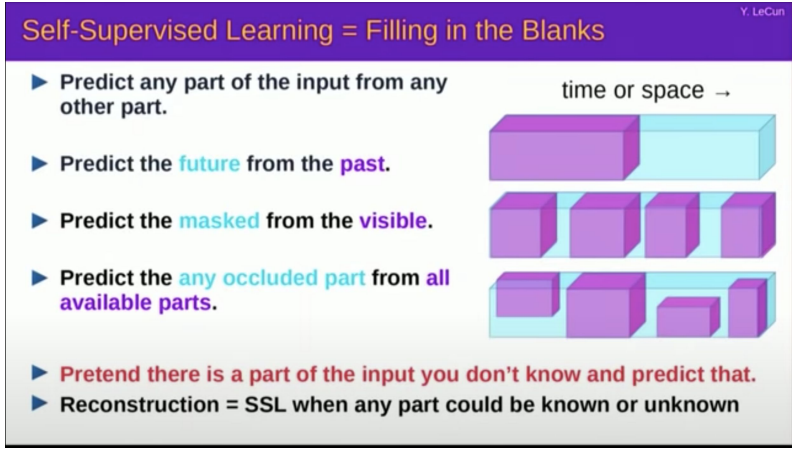
SAMSUNG Research



Self-Supervised Learning. Motivation.

- Learn the model of the world prior to learning a specific task
- Do this by predicting hidden parts of an input
- More bits of information per training sample

- Use less labeled data to train on a downstream task.

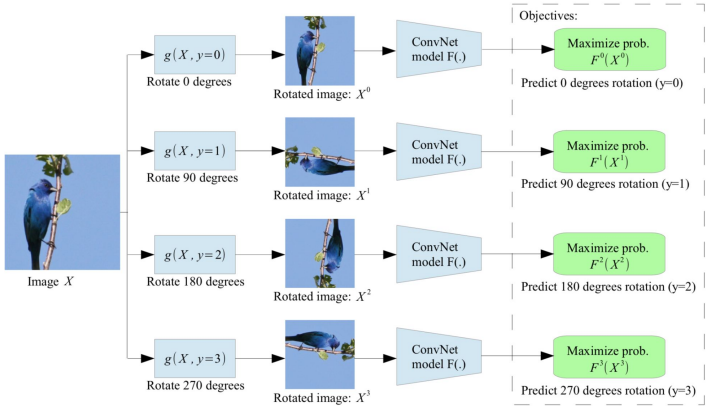


Yann Le Cunn, AAAI 2020 Keynotes Turing Award Winners Event

[https://www.youtube.com/watch?v=UX8OubxsY8w&t=2164s]

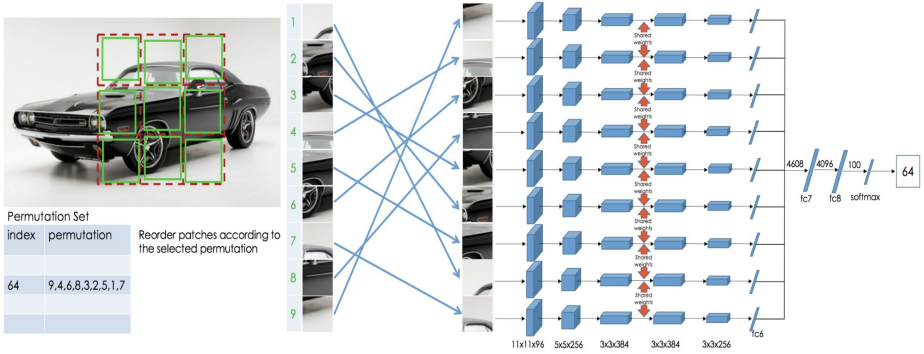
Pretext approaches

- Transform an image I with a random transformation
- Given I^t, predict the transformation t



Rotation

[https://arxiv.org/abs/1803.07728]

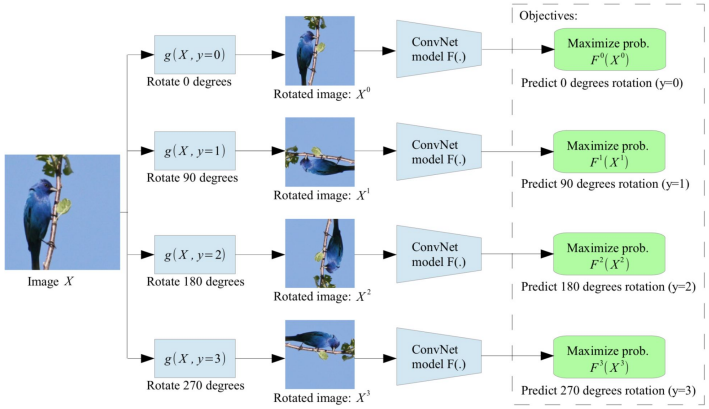


Jigsaw puzzle

[https://arxiv.org/abs/1603.09246]

Pretext approaches

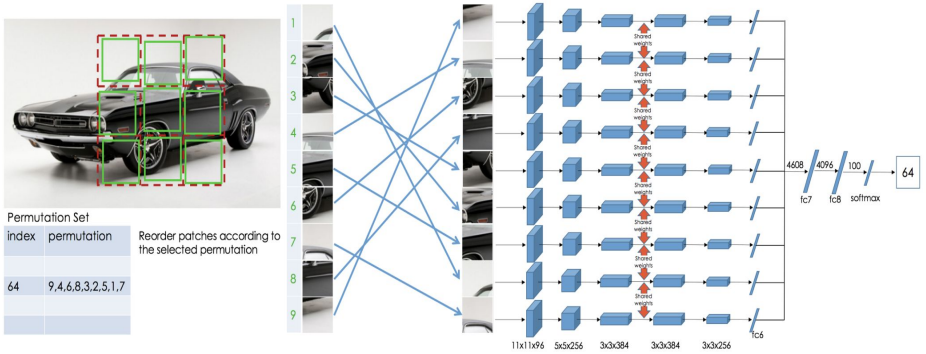
- Transform an image with a random transformation
- Given , predict the transformation t



Rotation

[https://arxiv.org/abs/1803.07728]

- Don’t learn representations directly.
- Want representations invariant to transformations



Jigsaw puzzle

[https://arxiv.org/abs/1603.09246]

Contrastive Prediction Coding

- Predict “future” patches of an image:

Problems:

- MSE loss doesn’t capture data variability.
- Learning generative model over pixels is time and computational consuming.
- Need to model local low-level information.

[https://arxiv.org/abs/1905.09272]

Contrastive Prediction Coding

- Encode patches into a vector space and predict next patch in this space

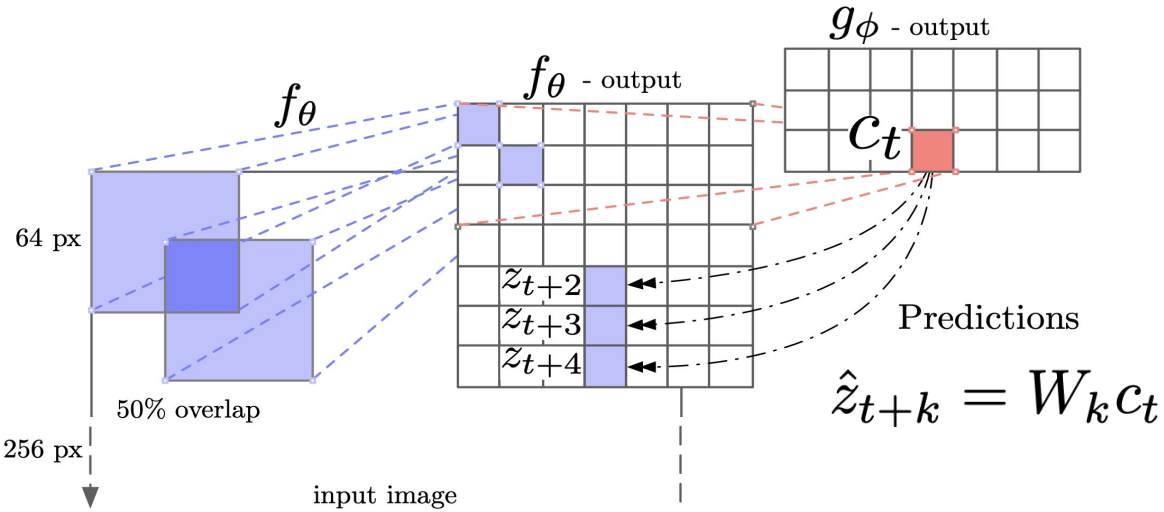


Figure 4: Visualization of Contrastive Predictive Coding for images (2D adaptation of Figure 1).

Contrastive Prediction Coding

- Encode patches into a vector space and predict next patch in this space

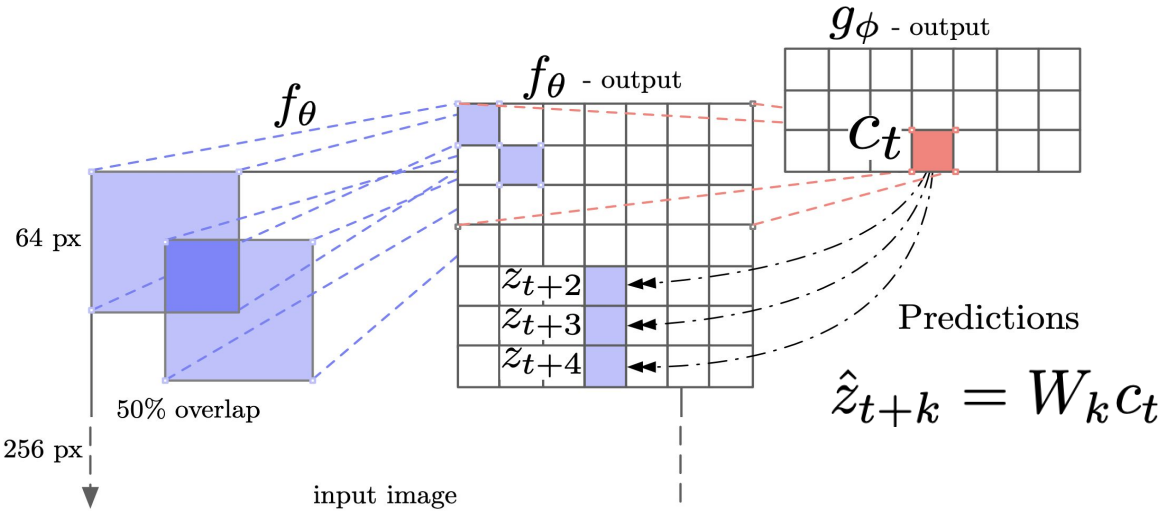


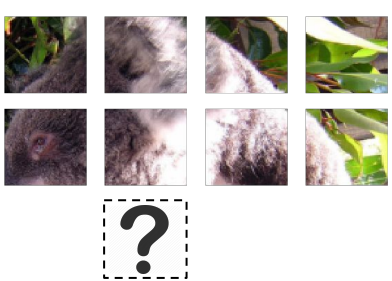
Figure 4: Visualization of Contrastive Predictive Coding for images (2D adaptation of Figure 1).

Problems with MSE?

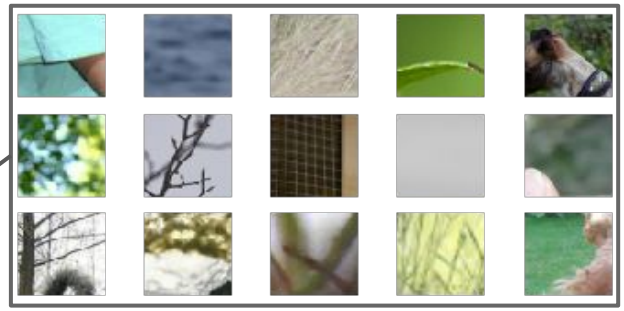
$$\|\hat{z}_t - z_t\|^2$$

Contrastive loss

- MSE loss $\left\|\hat{z}_t - z_t\right\|^2$ would lead to a constant prediction.
- Generative models are hard to train.
- Instead, use Noise Contrastive Estimation loss:



Positive sample

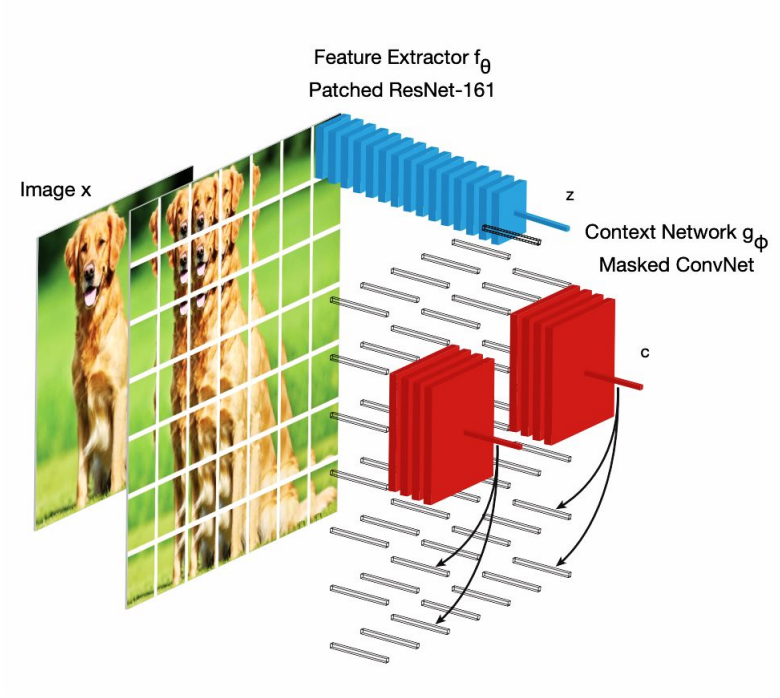


Negative samples

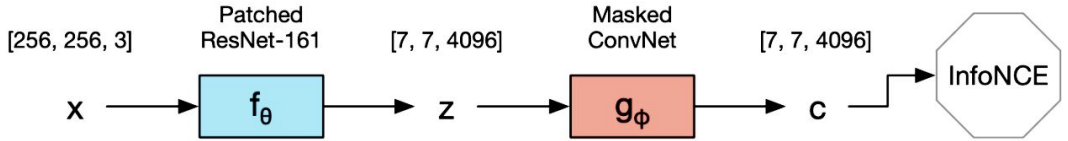
$$\log p(z_t \mid \hat{z}_t, \{z_l\}) = \log\left(\frac{\exp(\hat{z}_t^T z_t)}{\exp(\hat{z}_t^T z_t) + \sum_l \exp(\hat{z}_t^T z_l)}\right) \to \max$$

Predicted vector

Contrastive Prediction Coding



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



Pre-training

Linear Evaluation

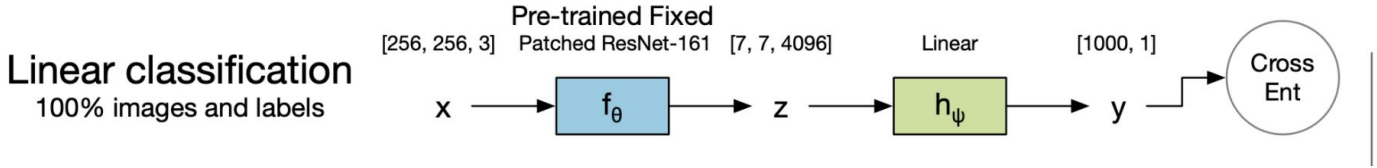


Table 1: Linear classification accuracy, and comparison to other self-supervised methods. In all cases the feature extractor is optimized in an unsupervised manner (using one of the methods listed below), and a linear classifier is trained on top using all labels in the ImageNet dataset.

Method	Architecture	Parameters (M)	Top-1	Top-5
Methods using ResNet-50:				
Local Aggregation [66]	ResNet-50	24	60.2	-
Momentum Contrast [25]	ResNet-50	24	60.6	-
CPC v2	ResNet-50	24	**63.8**	**85.3**
Methods using different architectures:				
Multi-task [13]	ResNet-101	28	-	69.3
Rotation [32]	RevNet-50 ×4	86	55.4	-
CPC v1 [58]	ResNet-101	28	48.7	73.6
BigBiGAN [15]	RevNet-50 ×4	86	61.3	81.9
AMDIM [5]	Custom-103	626	68.1	-
CMC [57]	ResNet-50 ×2	188	68.4	88.2
Momentum Contrast [25]	ResNet-50 ×4	375	68.6	-
CPC v2	ResNet-161	305	**71.5**	**90.1**

Linear Evaluation

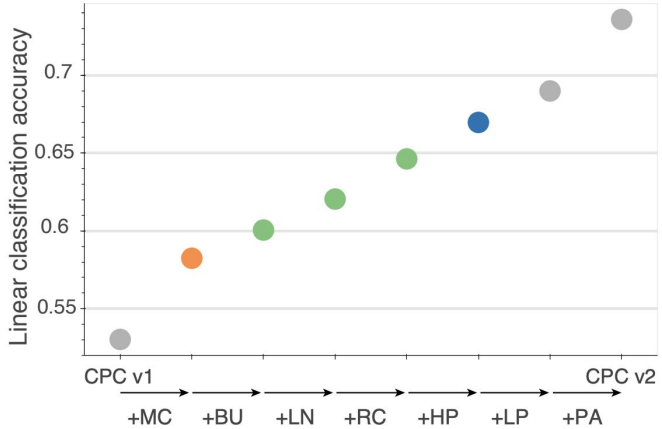
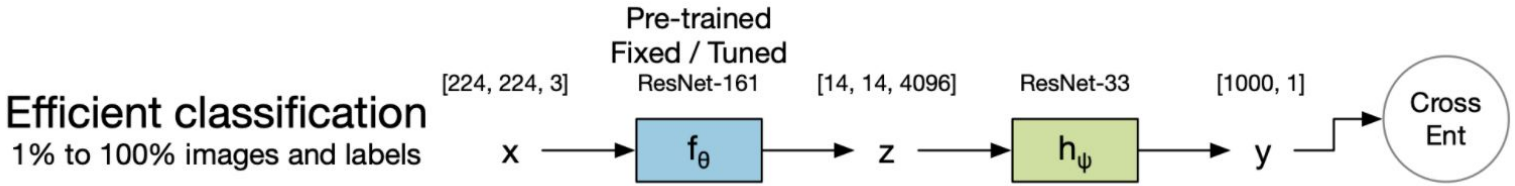
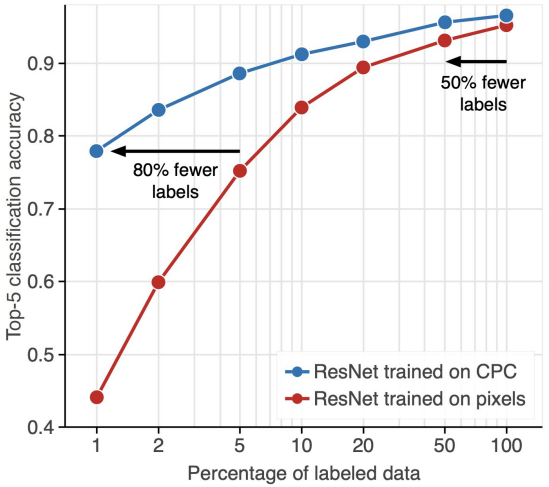


Figure 3: Linear classification performance of new variants of CPC, which incrementally add a series of modifications. MC: model capacity. BU: bottom-up spatial predictions. LN: layer normalization. RC: random color-dropping. HP: horizontal spatial predictions. LP: larger patches. PA: further patch-based augmentation. We use color to indicate the number of spatial predictions used (orange, green, blue for 1, 2 and 4 directions). Note that these accuracies are evaluated on a custom validation set and are therefore not directly comparable to the results we report and compare to.

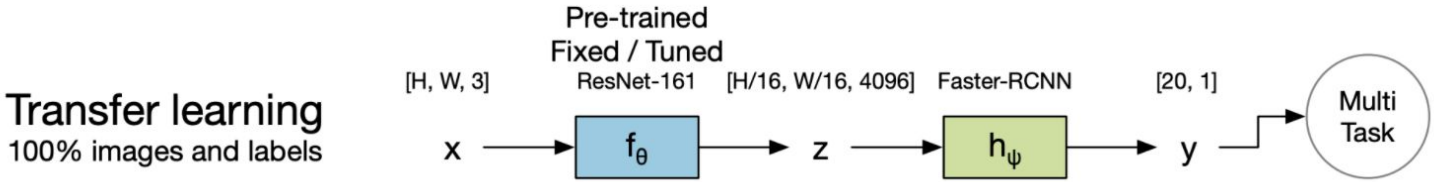
Semi-Supervised setting



Method	Architecture	Top-5 accuracy				
Labeled data		1%	5%	10%	50%	100%
†Supervised baseline	ResNet-200	44.1	75.2*	83.9	93.1	95.2#
Methods using label-propagation:						
Pseudolabeling [63]	ResNet-50	51.6	-	82.4	-	-
VAT + Entropy Minimization [63]	ResNet-50	47.0	-	83.4	-	-
Unsup. Data Augmentation [61]	ResNet-50	-	-	88.5	-	-
Rotation + VAT + Ent. Min. [63]	ResNet-50 ×4	-	-	**91.2**	-	95.0
Methods using representation learning only:						
Instance Discrimination [60]	ResNet-50	39.2	-	77.4	-	-
Rotation [63]	ResNet-152 ×2	57.5	-	86.4	-	-
ResNet on BigBiGAN (fixed)	RevNet-50 ×4	55.2	73.7	78.8	85.5	87.0
ResNet on AMDIM (fixed)	Custom-103	67.4	81.8	85.8	91.0	92.2
ResNet on CPC v2 (fixed)	ResNet-161	77.1	87.5	90.5	95.0	96.2
ResNet on CPC v2 (fine-tuned)	ResNet-161	**77.9***	**88.6**	**91.2**	**95.6#**	**96.5**



Transfer Learning



Method	Architecture	mAP
Transfer from labeled data:		
Supervised baseline	ResNet-152	74.7
Transfer from unlabeled data:		
Exemplar [17] by [13]	ResNet-101	60.9
Motion Segmentation [47] by [13]	ResNet-101	61.1
Colorization [64] by [13]	ResNet-101	65.5
Relative Position [14] by [13]	ResNet-101	66.8
Multi-task [13]	ResNet-101	70.5
Instance Discrimination [60]	ResNet-50	65.4
Deep Cluster [7]	VGG-16	65.9
Deeper Cluster [8]	VGG-16	67.8
Local Aggregation [66]	ResNet-50	69.1
Momentum Contrast [25]	ResNet-50	74.9
Faster-RCNN trained on CPC v2	ResNet-161	**76.6**

CPC in a nutshell

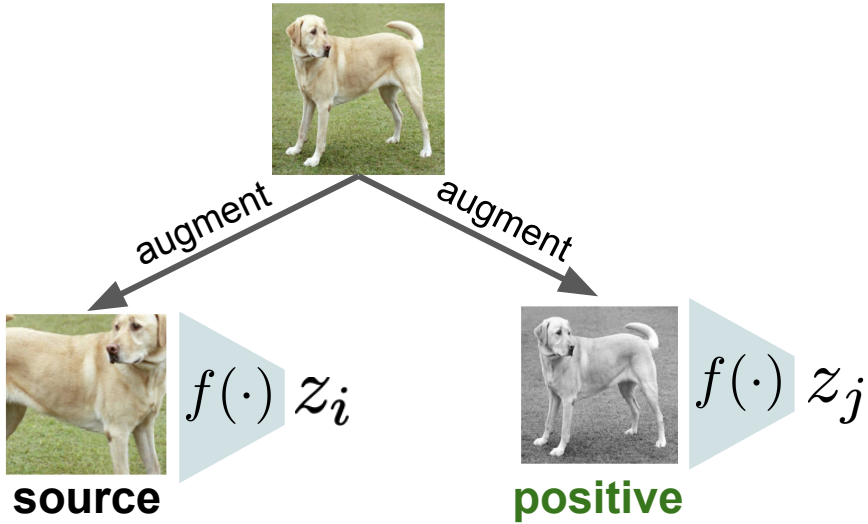
$$L_t = -\log\left(\frac{\exp(\hat{z}_t^T z_t)}{\exp(\hat{z}_t^T z_t) + \sum_l \exp(\hat{z}_t^T z_l)}\right)$$

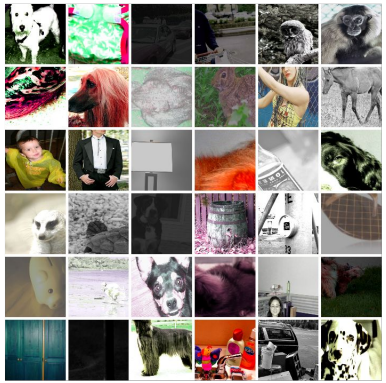
source positive negative

- **source** --- predicted embedding for a patch
- **positive** --- true embedding
- **negative** --- embeddings of random patches

Other (simple) ways to sample source, positive and negatives?

A Simple Framework for Contrastive Learning (SimCLR)





$$L_{i,j} = -\log\left(\frac{\exp(z_i^T z_j)}{\exp(z_i^T z_j) + \sum_l \exp(z_i^T z_l)}\right)$$

[https://arxiv.org/abs/2002.05709]

SimCLR

- Given an image find its augmentation among other images.
- Use the rest of batch as negative examples instead of a memory bank (need large batches, N=2-4K)

Batch of size N:

2N loss terms:

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

Data augmentation strategy

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

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

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)

Data augmentation strategy

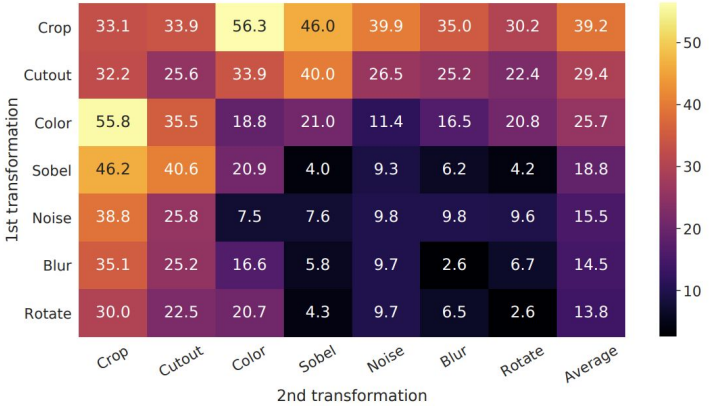
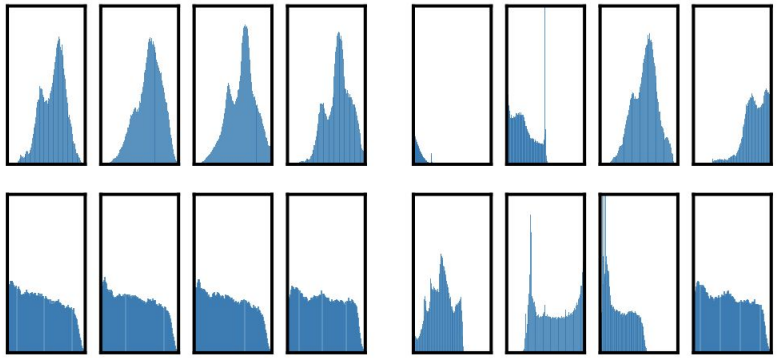


Figure 5. Linear evaluation (ImageNet top-1 accuracy) under individual or composition of data augmentations, applied only to one branch. For all columns but the last, diagonal entries correspond to single transformation, and off-diagonals correspond to composition of two transformations (applied sequentially).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



(a) Without color distortion. (b) With color distortion.

Figure 6. Histograms of pixel intensities (over all channels) for different crops of two different images (i.e. two rows). The image for the first row is from Figure 4. All axes have the same range.

A Simple Framework for Contrastive Learning

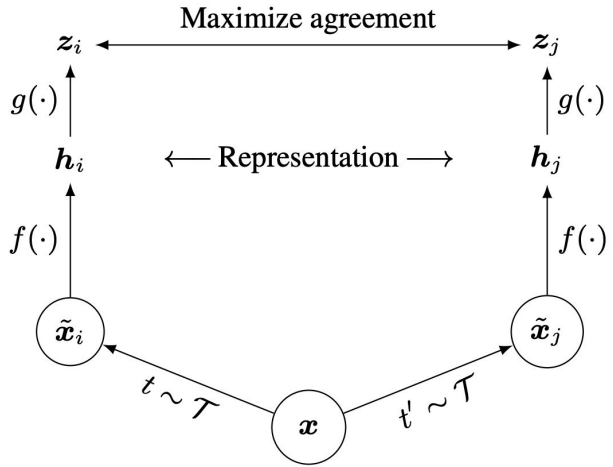


Figure 2. A simple framework for contrastive learning of visual representations. Two separate data augmentation operators are sampled from the same family of augmentations ($t \sim \mathcal{T}$ and $t' \sim \mathcal{T}$) and applied to each data example to obtain two correlated views. A base encoder network $f(\cdot)$ and a projection head $g(\cdot)$ are trained to maximize agreement using a contrastive loss. After training is completed, we throw away the projection head $g(\cdot)$ and use encoder $f(\cdot)$ and representation $\boldsymbol{h}$ for downstream tasks.

Projection head

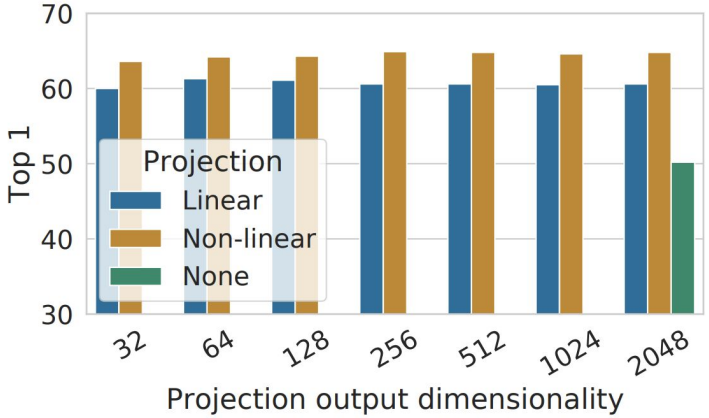


Figure 8. Linear evaluation of representations with different projection heads $g(\cdot)$ and various dimensions of $\boldsymbol{z} = g(\boldsymbol{h})$. The representation $\boldsymbol{h}$ (before projection) is 2048-dimensional here.

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

Table 3. Accuracy of training additional MLPs on different representations to predict the transformation applied. Other than crop and color augmentation, we additionally and independently add rotation (one of $\{0°, 90°, 180°, 270°\}$), Gaussian noise, and Sobel filtering transformation during the pretraining for the last three rows. Both $\boldsymbol{h}$ and $g(\boldsymbol{h})$ are of the same dimensionality, i.e. 2048.

Linear evaluation

Method	Architecture	Param.	Top 1	Top 5
Methods using ResNet-50:				
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**

Table 6. ImageNet accuracies of linear classifiers trained on representations learned with different self-supervised methods.

Semi-supervised evaluation

Method	Architecture	Label fraction 1%	Label fraction 10%
		Top 5	Top 5
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.

Conclusion

- Contrastive losses reduce the “object prediction” problem to classification
- The main difference is how to construct positive and negative samples
- Require large amount of negative samples and long training
  (best SimCLR takes 1.5 hours of 128 TPUs for 100 epochs)
- Contrastive self-supervised pretraining reaches (and outperforms) the level of supervised pretraining
- Performance saturated with increasing number of training examples