Representation learning with SSL models (with a probabilistic view)

TROPICAL PROBABILISTIC SUMMER SCHOOL

Diane Bouchacourt
Research Scientist, FAIR, Meta

Meta

FUNDAMENTAL AI RESEARCH AT METAFUNDAMENTAL AI RESEARCH AT META

Advance the state-of-the-art in artificial intelligence through open research for the benefit of all.

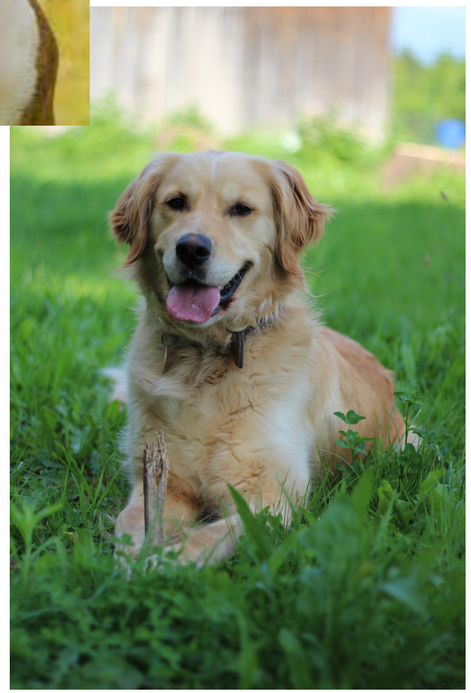
OUR TEAM GOAL

Building robust self-supervised architectures that understand the world and its variation, just like humans do.

“The key to artificial intelligence has always been the representation”

Jeff Hawkins, businessman, neuroscientist and engineer.

Constructing abstract representations of the world



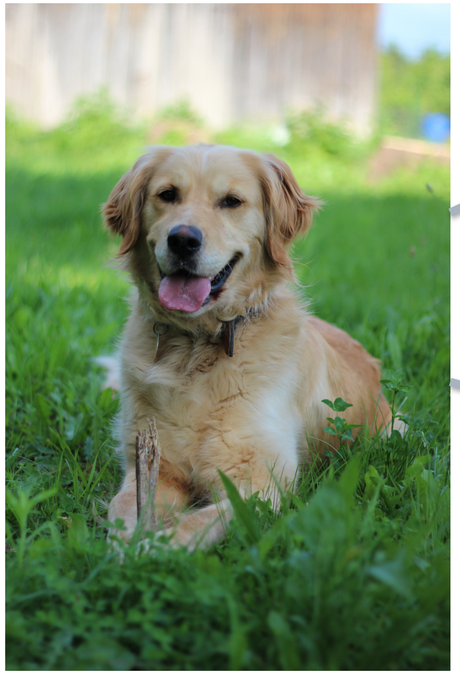
Breed

Pose

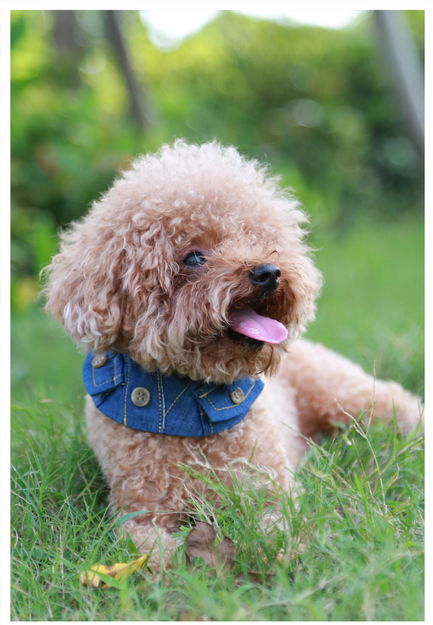
Color

...

Constructing abstract representations of the world allows generalization



Breed



Pose

01 What are desired (good) representation?

Representation learning

- Automatically *extracting meaningful patterns* from raw data
- Turning them into representation features that are easy to use for a variety of tasks e.g. classification, detection, interpretability, transfer learning.

Representation learning

- Automatically *extracting meaningful patterns* from raw data (e.g. factors of variations).
- Turning them into representation features that are easy to use for a variety of tasks e.g. classification, detection, interpretability, transfer learning.

What are *good features*?

- Features that allow to do a variety of tasks without retraining the representations.
- Features that capture high-level, semantic information and **generalize** to unseen settings.
- Features that are **robust to noise factors** that are not useful for downstream tasks.

This course

- A specific aspect of representation learning: self-supervised learning with InfoNCE-type losses.
- Self-supervised = no given “ground-truth labels” (i.e. unsupervised), but implicitly derives a target from unlabelled data.

02 Self-supervised representation learning with InfoNCE

Contrastive Predictive coding model

- Learning useful representation of input data by encoding shared, high-level information and discarding low-level information and noise.
- One of way to learn such representation in an unsupervised manner is to learn to **predict missing or future information *x***, from **context** *c* (e.g. past observations).
- Future information = implicit targets

Predicting future from context

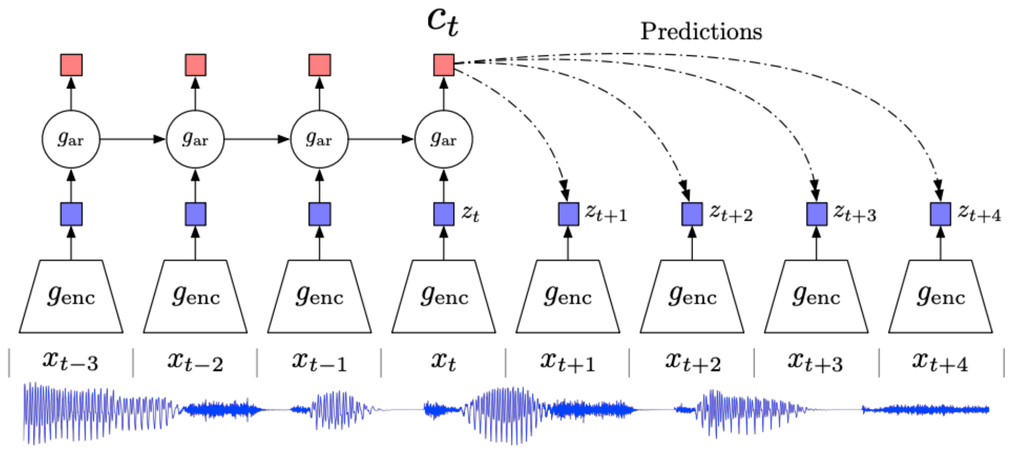


Figure 1: Overview of Contrastive Predictive Coding, the proposed representation learning approach. Although this figure shows audio as input, we use the same setup for images, text and reinforcement learning.

Should we just model *p(x|c)?*

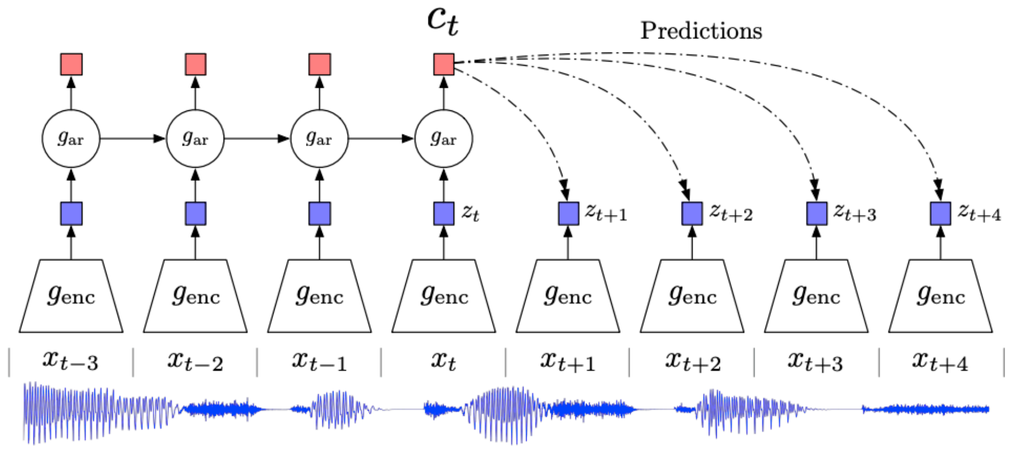


Figure 1: Overview of Contrastive Predictive Coding, the proposed representation learning approach. Although this figure shows audio as input, we use the same setup for images, text and reinforcement learning.

- Learning *p(x|c)* often leads to learning *complex, low-level information about x* that is not useful for the downstream tasks.
- Rather, Oord et al. propose to learn a compact representation that maximally optimizes mutual information between *x* and *c*

$$I(x;c) = \sum_{x,c} p(x,c) \log \frac{p(x \mid c)}{p(x)}$$

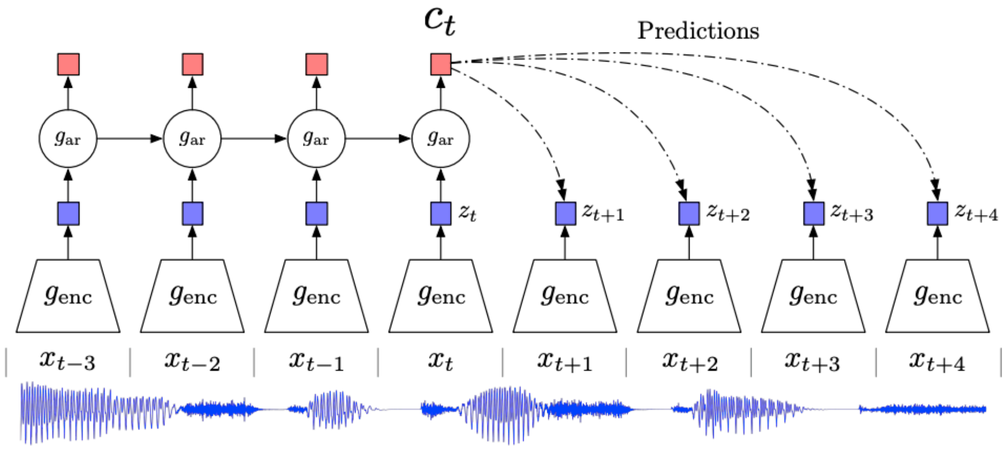


Figure 1: Overview of Contrastive Predictive Coding, the proposed representation learning approach. Although this figure shows audio as input, we use the same setup for images, text and reinforcement learning.

- An encoder model infers compact representations from the observations $g_{enc}(x_t) = z_t$
- An autoregressive model summarizes up previous encodings in a context latent representation $c_t = g_{ar}(z_{\leq t})$
- The model estimate a density ratio $f(x_{t+k}, c_t) \propto \dfrac{p(x_{t+k} \mid c_t)}{p(x_{t+k})}$

- The ratio is chosen to write as $f(x_{t+k}, c_t) = \exp(z_{t+k}^T W_k c_t) = \exp(z_{t+k}^T \hat{z}_{t+k})$
- How do we learn g, g_{ar}, W_k?
- For a given time step x_{t+k} , we have a set $X = \{x_1, \ldots x_N\}$ of N random samples containing:
    - 1 positive sample from $p(x_{t+k} \mid c_t)$
    - N-1 negative samples from a *proposal* distribution $p(x_{t+k})$

InfoNCE objective

$$\text{InfoNCE}_N = -\mathop{\mathbb{E}}_{X}\left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)}\right] = -\mathop{\mathbb{E}}_{X}\left[\log \frac{\exp(z_{t+k}^T \widehat{z}_{t+k})}{\sum_{x_j \in X} \exp(z_j^T \widehat{z}_{t+k})}\right]$$

Exercise: prove that minimizing InfoNCE_N results in $f(x_{t+k}, c_t) \propto \dfrac{p(x_{t+k} \mid c_t)}{p(x_{t+k})}$

Exercise

$$\text{InfoNCE}_N = -\mathop{\mathbb{E}}_{X}\left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)}\right] = -\mathop{\mathbb{E}}_{X}\left[\log \frac{\exp(z_{t+k}^T \widehat{z}_{t+k})}{\sum_{x_j \in X} \exp(z_j^T \widehat{z}_{t+k})}\right]$$

Hint: this looks familiar… It is the cross-entropy loss for a N-way classifier: **finding the positive sample among the X possible candidates.**

Exercise

Noise contrastive: distinguish the true sample from $p(x_{t+k} \mid c_t)$ from the “fake” ones, coming from a *noise* distribution, here the marginal $p(x_{t+k})$. Let us write the event that x_i is the positive sample as $[d = i]$.

- Model’s probability

$$q(d = i \mid X, c_t) = \frac{f_k(x_i, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)}$$

- What is the true $p(d = i \mid X, c_t)$?

Exercise

Sampling process: draw uniformly $i \sim [1,...,N]$

For each $l = 1,...,N$, if $l = i$ then $x_l \sim p(x_{t+k} \mid c_t)$ else $x_l \sim p(x_{t+k})$

Thus, $p(d = i, X, c_t) = \frac{1}{N} p(x_i \mid c_t) \prod_{l \neq i} p(x_l)$

Exercise

Sampling process: draw uniformly $i \sim [1,...,N]$

For each $l = 1,...,N$, if $l = i$ then $x_l \sim p(x_{t+k} \mid c_t)$ else $x_l \sim p(x_{t+k})$

Thus, $p(d = i, X, c_t) = \frac{1}{N} p(x_i \mid c_t) \prod_{l \neq i} p(x_l)$

And the conditional writes as

$$p(d = i \mid X, c_t) = \frac{p(d = i \mid X, c_t)}{p(X, c_t)} = \frac{\frac{1}{N} p(x_i \mid c_t) \prod_{l \neq i} p(x_l)}{p(X, c_t)} = \frac{\frac{1}{N} p(x_i \mid c_t) \prod_{l \neq i} p(x_l)}{\sum_{j=1}^{N} p(d = j, X, c_t)} = \frac{p(x_i \mid c_t) \prod_{l \neq i} p(x_l)}{\sum_{j=1}^{N} p(x_j \mid c_t) \prod_{l \neq j} p(x_l)} = \frac{\frac{p(x_i \mid c_t)}{p(x_i)}}{\sum_{j=1}^{N} \frac{p(x_j \mid c_t)}{p(x_j)}}.$$

Exercise

So we see that if InfoNCE$_N$ is minimized, we have the model probability

$q(d = i \mid X, c_t) = \dfrac{f_k(x_i, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)}$ equal to the true probability $p(d = i \mid X, c_t) = \dfrac{\dfrac{p(x_i \mid c_t)}{p(x_i)}}{\sum_{j=1}^{N} \dfrac{p(x_j \mid c_t)}{p(x_j)}}$

Thus, $f(x_{t+k}, c_t) \propto \dfrac{p(x_{t+k} \mid c_t)}{p(x_{t+k})}$

InfoNCE as a lower bound on Mutual Information

Recall, Oord et al. propose to learn a compact representation that maximizes mutual information between x and c.

InfoNCE as a lower bound on Mutual Information

Recall, Oord et al. propose to learn a compact representation that maximizes mutual information between *x* and *c.* The MI between the context and the future information writes as follows (see Oord et al.):

$$I(x_{t+k}, c_t) \geq \log(N) - \text{InfoNCE}_N,$$

which becomes tighter as N becomes larger.

Application of CPC to speaker representation learning and classification

From representation learning unsupervised to supervised classification:

Once the unsupervised representation pre-training is done, train a **supervised linear head** on top of the frozen representation encoder (or fine-tuning of the whole model).

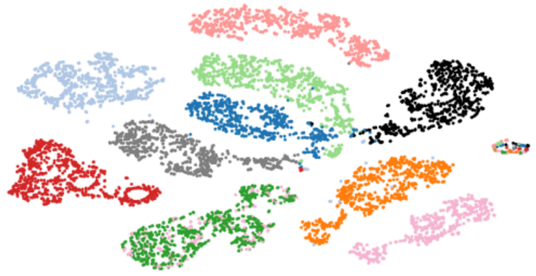


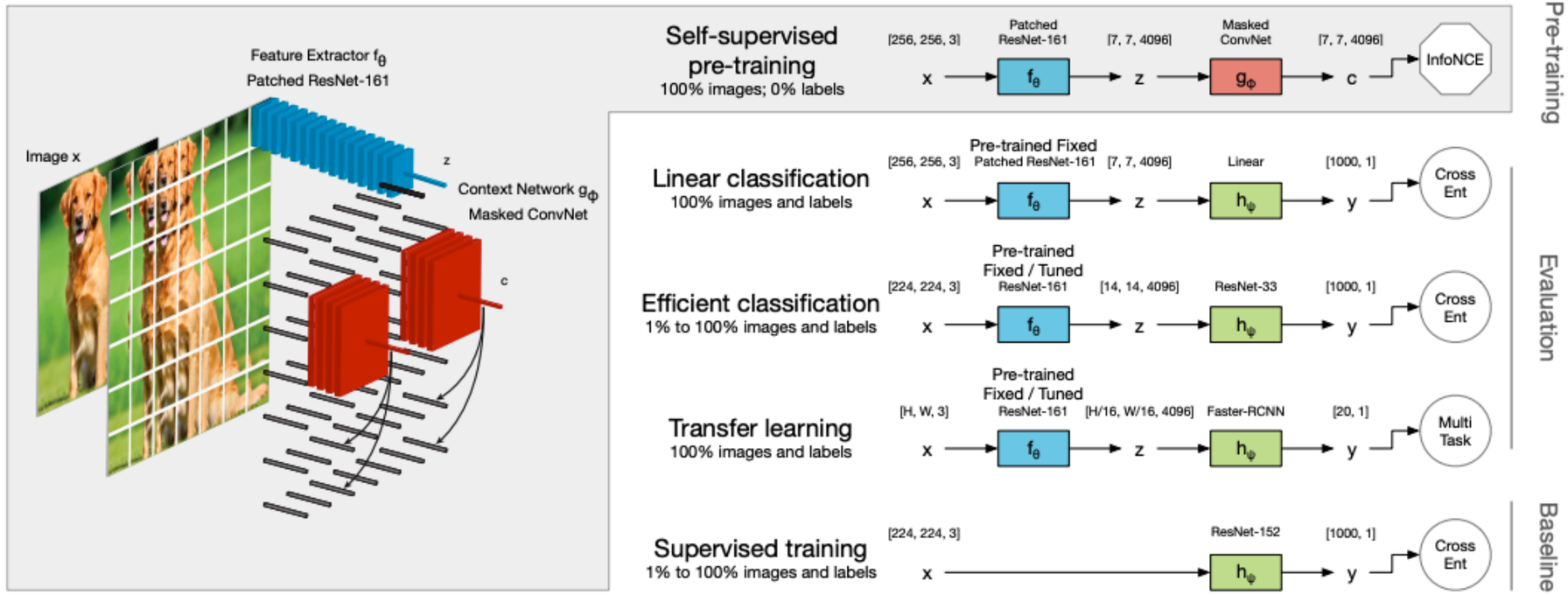
Figure 2: t-SNE visualization of audio (speech) representations for a subset of 10 speakers (out of 251). Every color represents a different speaker.

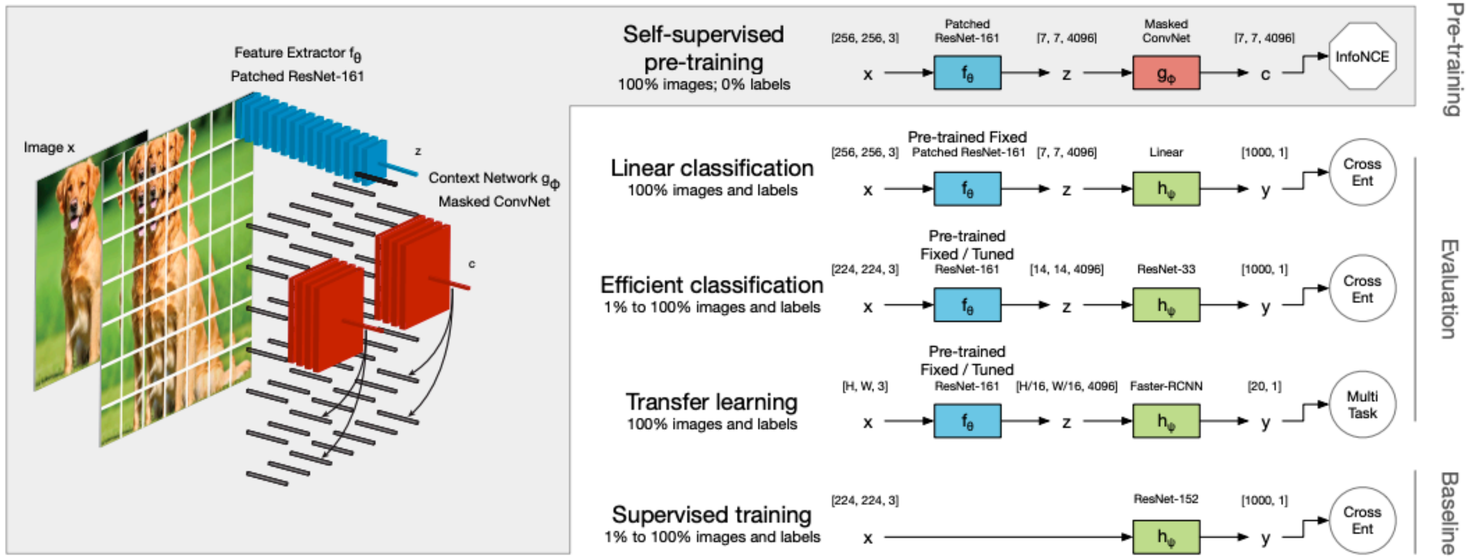
Method	ACC
Phone classification	
Random initialization	27.6
MFCC features	39.7
CPC	64.6
Supervised	74.6
Speaker classification	
Random initialization	1.87
MFCC features	17.6
CPC	97.4
Supervised	98.5

Table 1: LibriSpeech phone and speaker classification results. For phone classification there are 41 possible classes and for speaker classification 251. All models used the same architecture and the same audio input sizes.

03 SSL pre-training for image classification

Self-supervised pre-trained representations in image classification [Hénaff et al.]





- Each input image is first divided into a set of overlapping patches each of which is encoded with a neural network into a single feature vector $z_{i,j}$.
- Similar to the previous example, the prediction task then consists of predicting “future” vectors $z_{i+k,j}$ from current context vectors $c_{i,j}$.
- Very good results on ImageNet + transfer learning on Pascal VOC.

SimCLR Chen et al.

- Adapts InfoNCE loss in order to learn useful representation of image data.
- Maximize agreement between two representations from two different crops of the same image.

(click on GIF below)

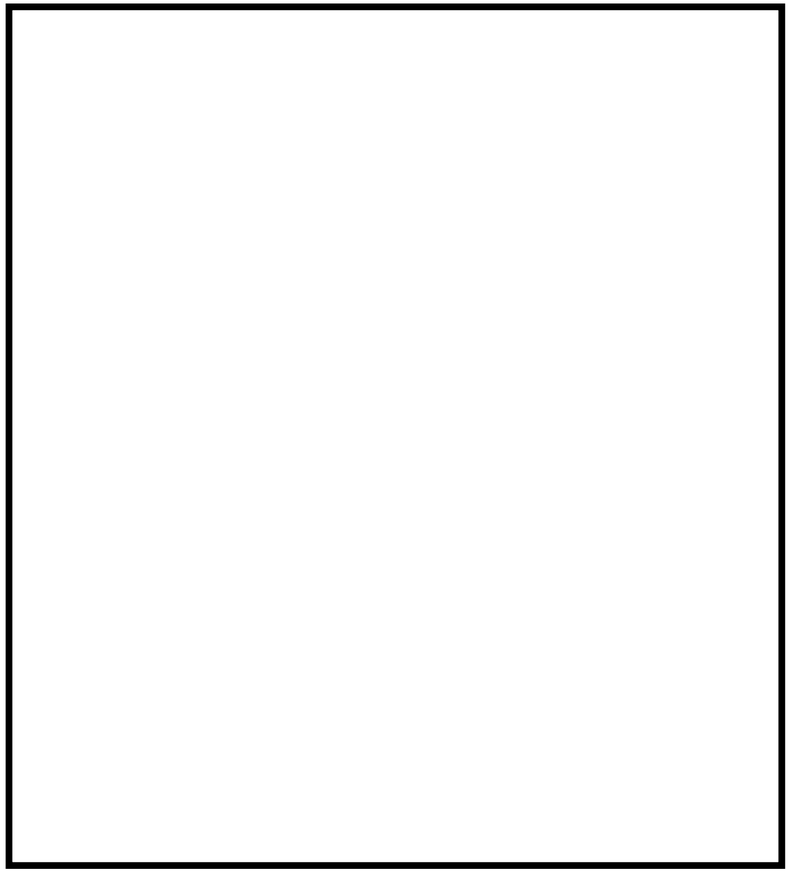


Illustration adapted from the author blog post https://blog.research.google/2020/04/advancing-self-supervised-and-semi.html

SimCLR Chen et al.

- Minibatch of N images each augmented 2 ways, resulting in 2N data points $\tilde{x}_i$.
- CNN model (e.g. ResNet) produces representations $h_i = f(\tilde{x}_i)$
- These are fed to a projection network (MLP) producing $z_i = g(h_i)$ (lower dimensionality)

(click on GIF below)

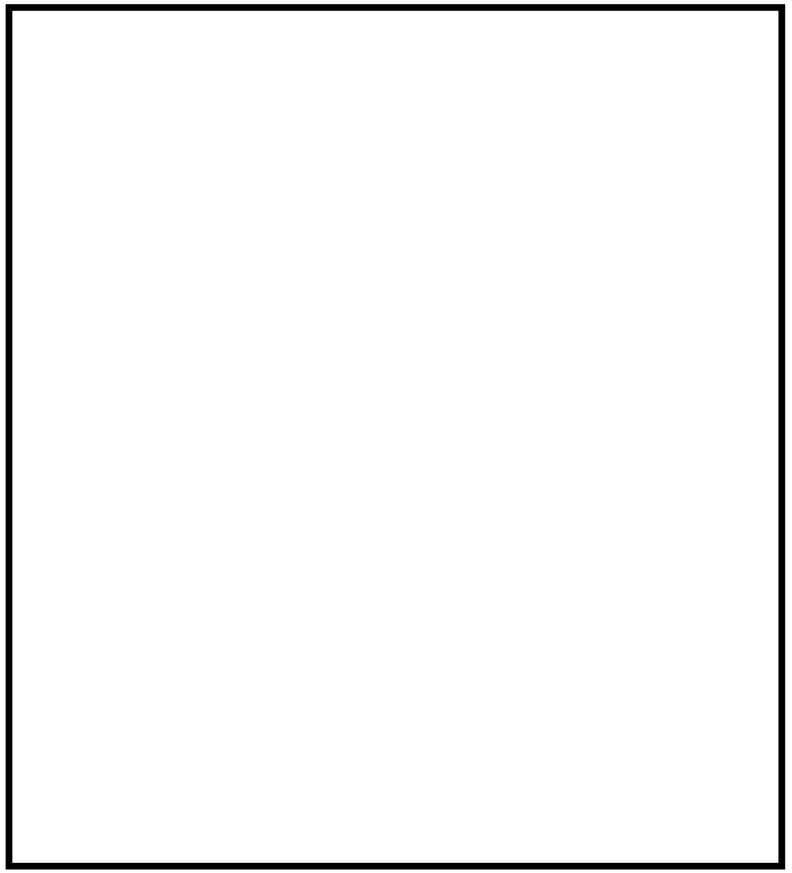
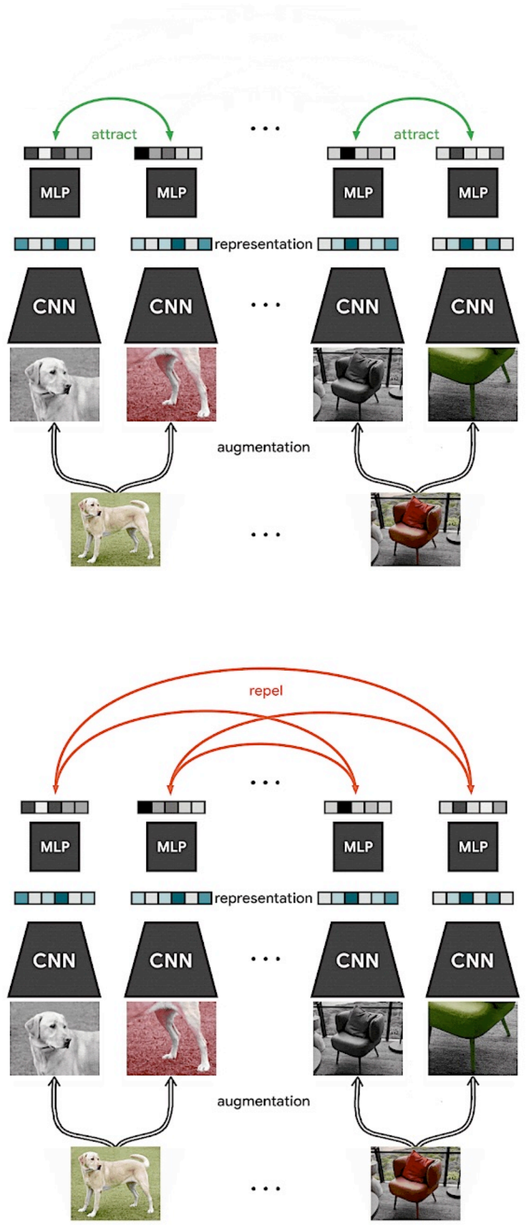


Illustration adapted from the author blog post https://blog.research.google/2020/04/advancing-self-supervised-and-semi.html

SimCLR [Chen et al.]

- Positives are crops from the same image, while negatives are the other 2(N-1) augmented samples within a minibatch.
- InfoNCE loss is used to push samples from the same image closer, while pushing away all other augmented samples (negatives):

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



A revolution in image classification

- Results are close to fully supervised architectures on ImageNet.
- Allows to learn from *wide corpus of unlabelled data.*
- Ability to transfer learning (pre-trained solely on ImageNet).

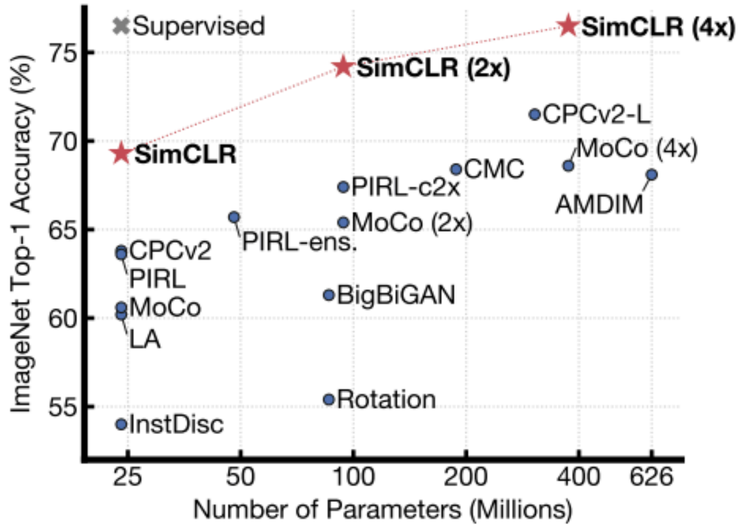


Figure 1. ImageNet Top-1 accuracy of linear classifiers trained on representations learned with different self-supervised methods (pretrained on ImageNet). Gray cross indicates supervised ResNet-50. Our method, SimCLR, is shown in bold.

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

Table 8. Comparison of transfer learning performance of our self-supervised approach with supervised baselines across 12 natural image classification datasets, for ResNet-50 $(4\times)$ models pretrained on ImageNet. Results not significantly worse than the best ($p > 0.05$, permutation test) are shown in bold. See Appendix B.8 for experimental details and results with standard ResNet-50.

Figures from Wang and Isola

Causal interpretation by Kügelgen et al

- Consider a latent variable model by assuming an embedding (latent) space (z space) that partitions into two parts:
  - **content** (e.g. class of the object) which is untouched by augmentation
  - **style**, which is allowed to change
  - Augmentations = intervention on the style component

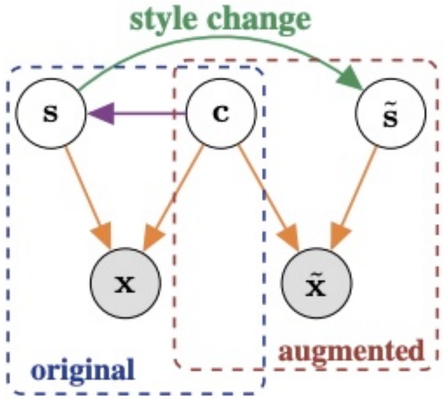


Figure 1: **Overview of our problem formulation.** We partition the latent variable $\mathbf{z}$ into content $\mathbf{c}$ and style $\mathbf{s}$, and allow for statistical and causal dependence of style on content. We assume that only style changes between the original view $\mathbf{x}$ and the augmented view $\tilde{\mathbf{x}}$, i.e., they are obtained by applying the same deterministic function $\mathbf{f}$ to $\mathbf{z} = (\mathbf{c}, \mathbf{s})$ and $\tilde{\mathbf{z}} = (\mathbf{c}, \tilde{\mathbf{s}})$.

Causal interpretation by Kügelgen et al

- Consider a latent variable model by assuming an embedding (latent) space (z space) that partitions into two parts:
  - **content** (e.g. class of the object) which is untouched by augmentation
  - **style**, which is allowed to change
- Augmentations = soft style intervention

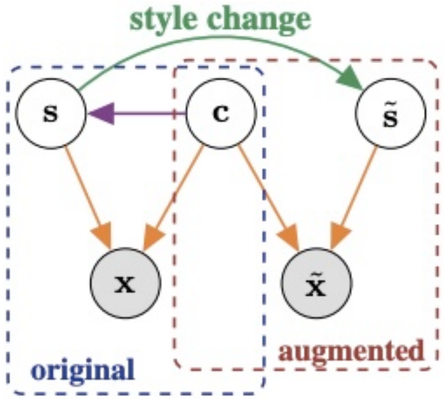


Figure 1: **Overview of our problem formulation.** We partition the latent variable $\mathbf{z}$ into content $\mathbf{c}$ and style $\mathbf{s}$, and allow for statistical and causal dependence of style on content. We assume that only style changes between the original view $\mathbf{x}$ and the augmented view $\tilde{\mathbf{x}}$, i.e., they are obtained by applying the same deterministic function $\mathbf{f}$ to $\mathbf{z} = (\mathbf{c}, \mathbf{s})$ and $\tilde{\mathbf{z}} = (\mathbf{c}, \tilde{\mathbf{s}})$.

Causal interpretation by Kügelgen et al

- With this model, Kügelgen et al show that SSL with data augmentations as in SimCLR **provably separates the content component from the style component.**
- Therefore, if we consider classification, it is really easier for a linear head on top of the representation to extract only the content, i.e. the class.

- Our work Eastwood et al. extends it to show that every component of the style part (e.g. color, color) can similarly be decomposed in the latent space.

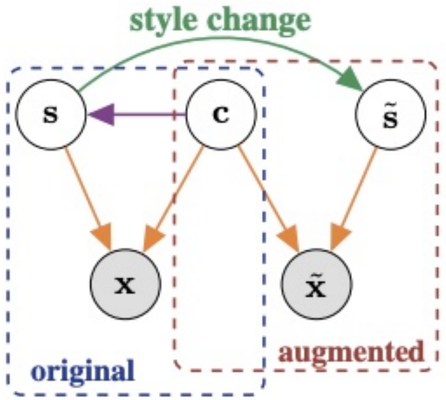


Figure 1: **Overview of our problem formulation.** We partition the latent variable $\mathbf{z}$ into content $\mathbf{c}$ and style $\mathbf{s}$, and allow for statistical and causal dependence of style on content. We assume that only style changes between the original view $\mathbf{x}$ and the augmented view $\tilde{\mathbf{x}}$, i.e., they are obtained by applying the same deterministic function $\mathbf{f}$ to $\mathbf{z} = (\mathbf{c}, \mathbf{s})$ and $\tilde{\mathbf{z}} = (\mathbf{c}, \tilde{\mathbf{s}})$.

General form

- p_{data} the data distribution
- p_{pos} the distribution of positive pairs
- f an encoder model mapping data to normalized feature vectors
- M number of negatives samples

$$\text{InfoNCE}(f, \tau, M) = \mathop{\mathbb{E}}_{(x,y)\sim p_{pos}, \{x_k^-\}_{k=1}^{M}\sim p_{data}} \left[-\log \frac{\exp(f(x)^T f(y)/\tau)}{\exp(f(x)^T f(y))/\tau) + \sum_{k=1}^{N} \exp(f(x)^T f(x_k^-)/\tau)} \right]$$

04 SSL through Alignment and Uniformity

Alignment & uniformity

The interpretation of InfoNCE as a mutual information lower bound is inconsistent to explain its success: maximizing lower bounds on MI can result in bad representations while looser bounds can lead to better representations (see Tschannen et al.).

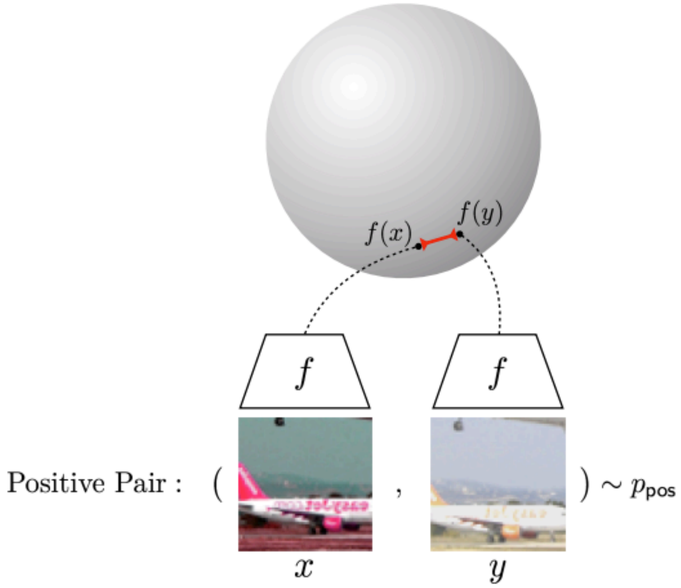
Alignment & uniformity

The interpretation of InfoNCE as a mutual information lower bound is inconsistent to explain its success: maximizing lower bounds on MI can result in bad representations while looser bounds can lead to better representations (see Tschannen et al.).

Instead, Wang et Isola propose to explain the performances of InfoNCE with the principles of *alignment and uniformity.*

Alignment & uniformity

Alignment pushes similar samples to have similar features and be close, *uniformity* preserves as much information as possible about the data by distributing representation vectors uniformly.



Alignment: Similar samples have similar features.
(Figure inspired by Tian et al. (2019).)

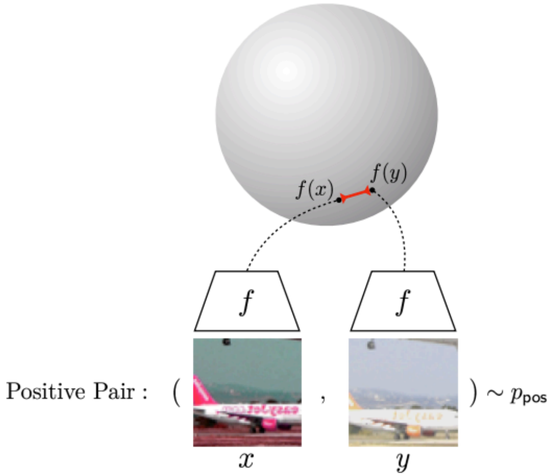
Uniformity: Preserve maximal information.

Figure 1: Illustration of alignment and uniformity of feature distributions on the output unit hypersphere. STL-10 (Coates et al., 2011) images are used for demonstration.

Figures from Wang and Isola

Alignment & uniformity

$$\lim_{M\to\infty} L(f,\tau,M) - \log(M) = \underbrace{\mathop{\mathbb{E}}_{(x,y)\sim p_{pos}} \left[-f(x)^T f(y)/\tau\right]}_{\text{Alignment}} + \underbrace{\mathop{\mathbb{E}}_{x\sim p_{data}} \left[-\log \mathop{\mathbb{E}}_{x^-\sim p_{data}} \left[\exp(f(x)^T f(x^-))/\tau\right]\right]}_{\text{Uniformity}}$$



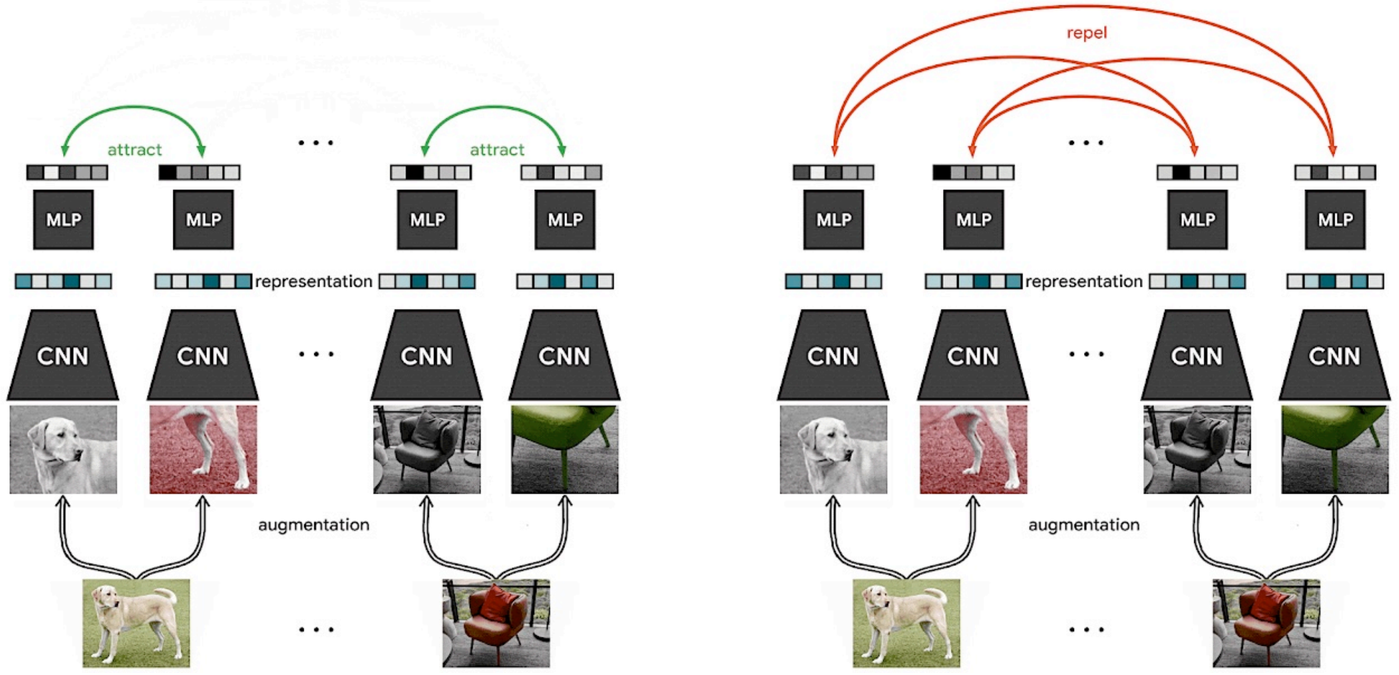
Alignment: Similar samples have similar features. (Figure inspired by Tian et al. (2019).)

Uniformity: Preserve maximal information.

Figure 1: Illustration of alignment and uniformity of feature distributions on the output unit hypersphere. STL-10 (Coates et al., 2011) images are used for demonstration.

Alignment & uniformity

$$\lim_{M\to\infty} L(f,\tau,M) - \log(M) = \underbrace{\mathop{\mathbb{E}}_{(x,y)\sim p_{pos}} \left[-f(x)^T f(y)/\tau\right]}_{\text{Alignment}} + \underbrace{\mathop{\mathbb{E}}_{x\sim p_{data}} \left[-\log \mathop{\mathbb{E}}_{x^-\sim p_{data}} \left[\exp(f(x)^T f(x^-))/\tau\right]\right]}_{\text{Uniformity}}$$



Why does the hypersphere matter?

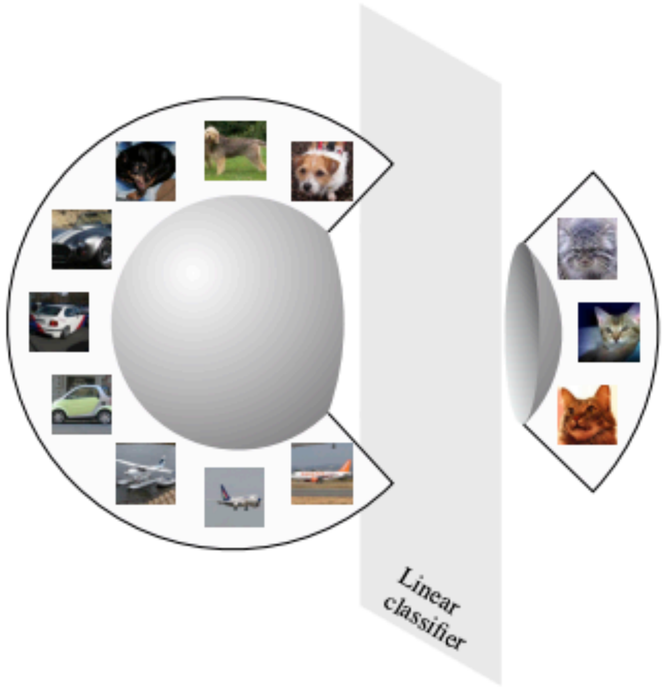
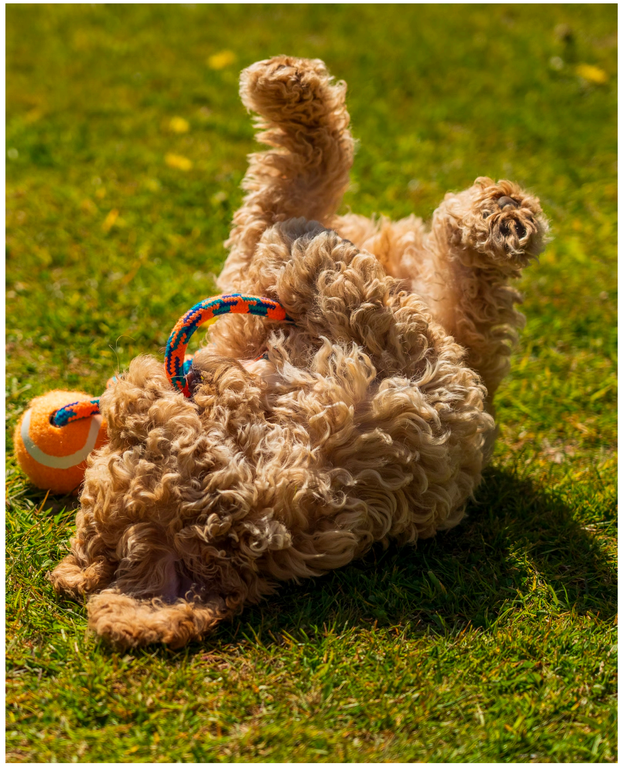


Figure 2: **Hypersphere:** When classes are well-clustered (forming spherical caps), they are linearly separable. The same does not hold for Euclidean spaces.

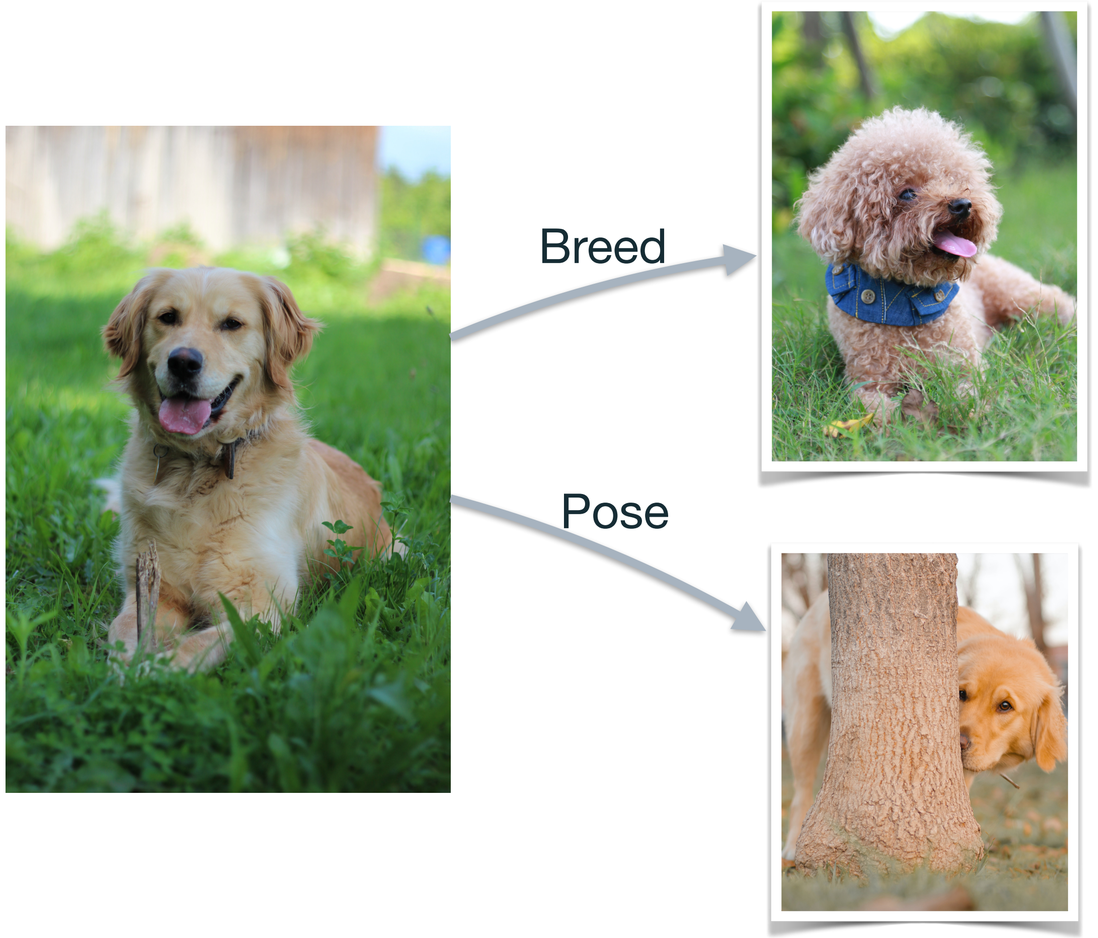
Figures from Wang and Isola

05 Robust self-supervised learning with Lie groups

Do they generalize?

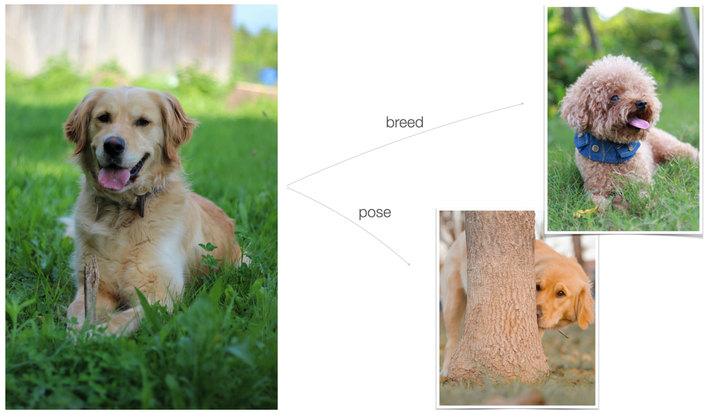


Do they generalize?



05 ROBUST SELF-SUPERVISED LEARNING WITH LIE GROUPS

Training

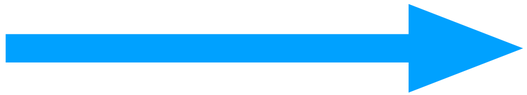


Test

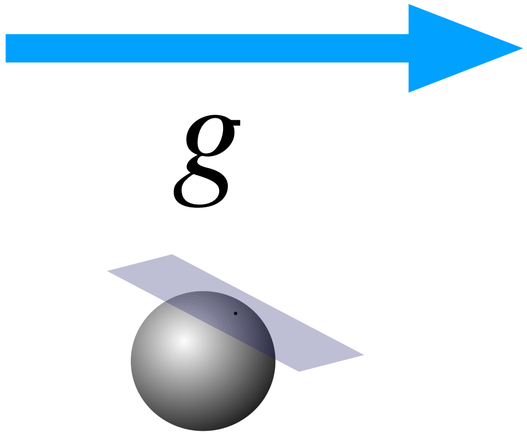
Gen alize

"The robustness limits of sota vision models to natural variation" M Ibrahim, Q Garrido, A Morcos*, D Bouchacourt* 2021

Learning natural variations from data

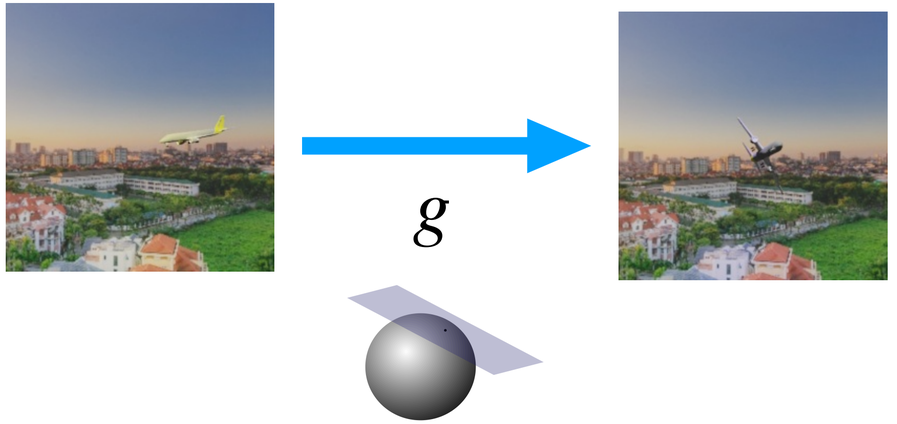


Learning natural variations from data



Lie algebra

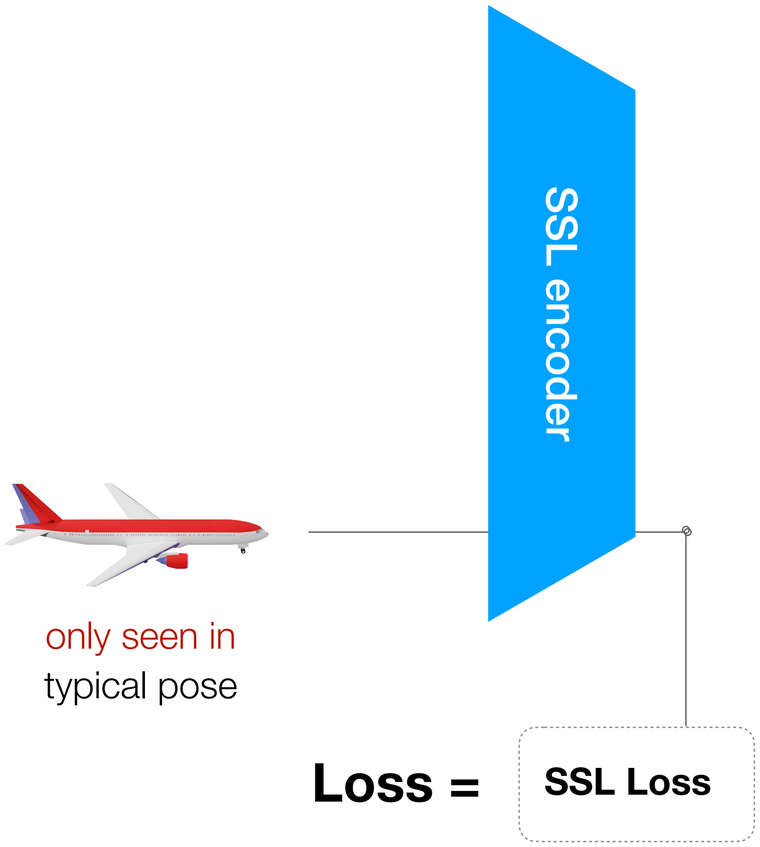
Learning natural variations from data



- **Lie groups** are continuous groups described by a set of real parameters (see Hall, 2003).
- Many continuous transformations (e.g., rotations) are Lie groups, but Lie groups lack the typical structure of a vector space.
- Lie group have a corresponding Lie algebra: a vector space that can be described using basis matrices, allowing to describe the infinite number elements of the group by a finite number of basis matrices.
- *Learn these basis matrices to model the data variations in the representations.*

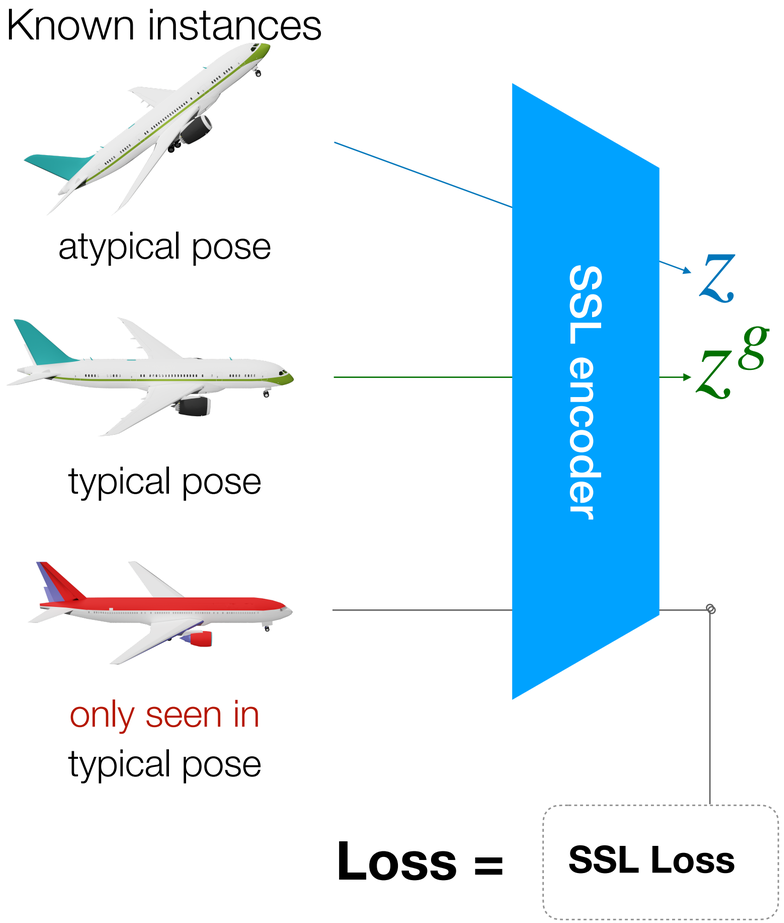
Training Data

Known instances



Natural variation

Training Data



Natural variation

Training Data

Self-Supervised Lie Operator

Known instances

atypical pose

typical pose

only seen in typical pose

SSL encoder

z

z^g

$\hat{t}$

infer transformation magnitude

encourages

$$\hat{g}(z) = z^g$$

Lie Operator

$$\hat{g} = exp(\sum_d \hat{t}_d L_d)$$

learned Lie algebra basis

Loss = **SSL Loss** + **Contrastive Lie Loss**

Natural variation

Training Data

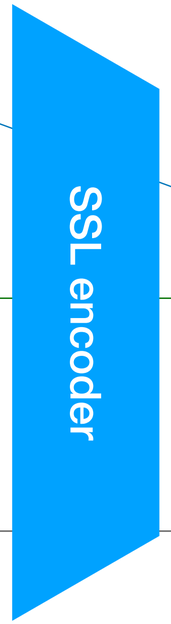
Self-Supervised Lie Operator

Known instances

atypical pose

typical pose

only seen in typical pose



z

z^g

$\hat{t}$

infer transformation magnitude

encourages

$$\hat{g}(z) = z^g$$

Lie Operator

$$\hat{g} = exp(\sum_d \hat{t}_d L_d)$$

learned Lie algebra basis

Loss = **SSL Loss** + **Contrastive Lie Loss** + $\|\hat{g}(z) - z^g\|_2^2$

Natural variation

Structure variation as Transformations

in representation space

Training Data

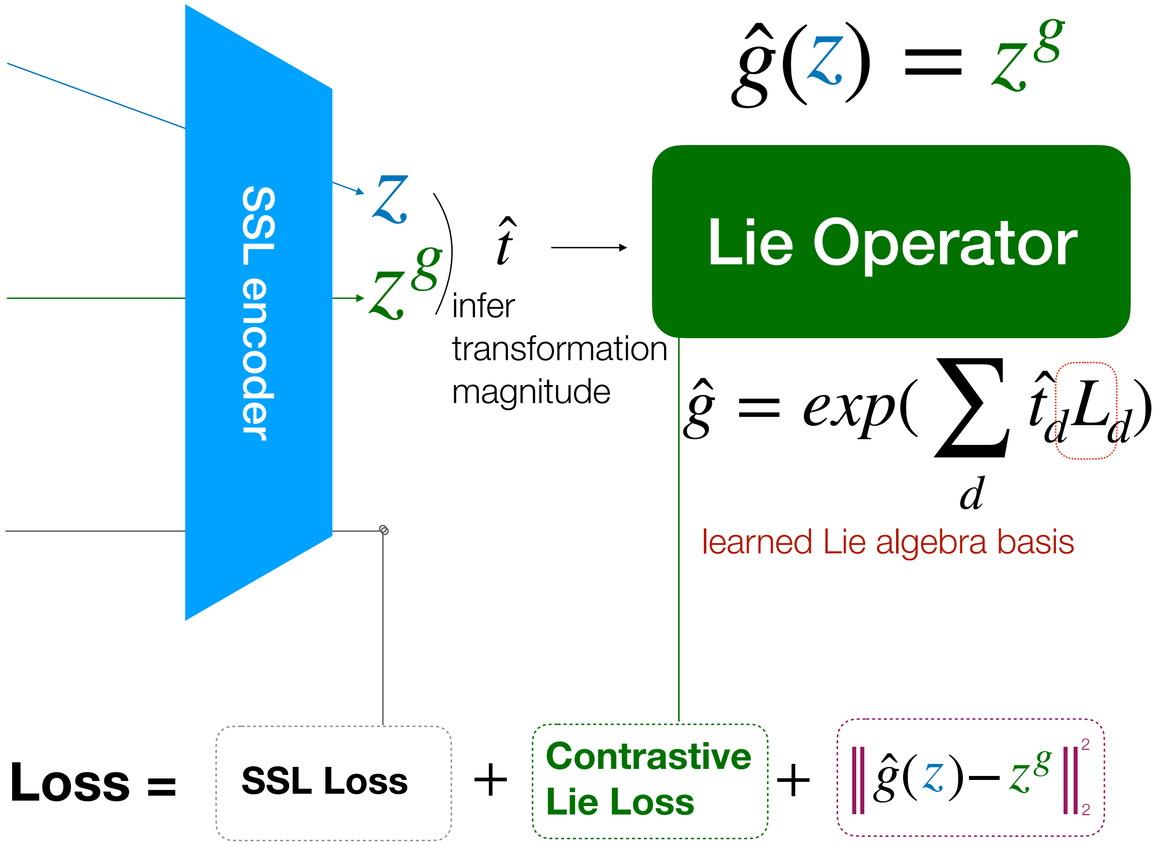
Known instances

atypical pose

typical pose

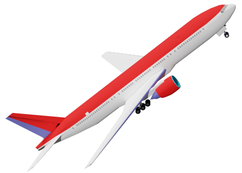
only seen in typical pose

Self-Supervised Lie Operator



Evaluate Robustness

MAE Lie



Known instance in new pose **+13.7%**

Unknown instance **+10.3%**

Unknown instance in typical pose **+12.4%**

Natural variation

Structure variation as Transformations

in representation space

Generalize variation across instances

Training Data

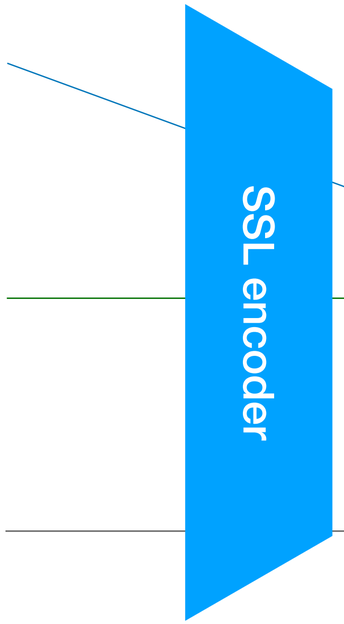
Known instances

atypical pose

typical pose

only seen in typical pose

Self-Supervised Lie Operator



z

z^g

$\hat{t}$

infer transformation magnitude

encourages

$$\hat{g}(z) = z^g$$

Lie Operator

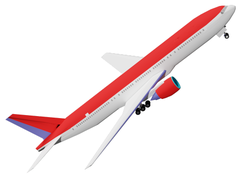
$$\hat{g} = exp(\sum_d \hat{t}_d L_d)$$

learned Lie algebra basis

Loss = **SSL Loss** + **Contrastive Lie Loss** + $\|\hat{g}(z) - z^g\|_2^2$

Evaluate Robustness

VICReg



Known instance in new pose — **+7.7%**

Unknown instance — **+1.2%**

Unknown instance in typical pose — **+0.2%**

Natural variation

Structure variation as Transformations

in representation space

Generalize variation across instances

06 Open-vocabulary classification with Vision-Language Models (VLMs)

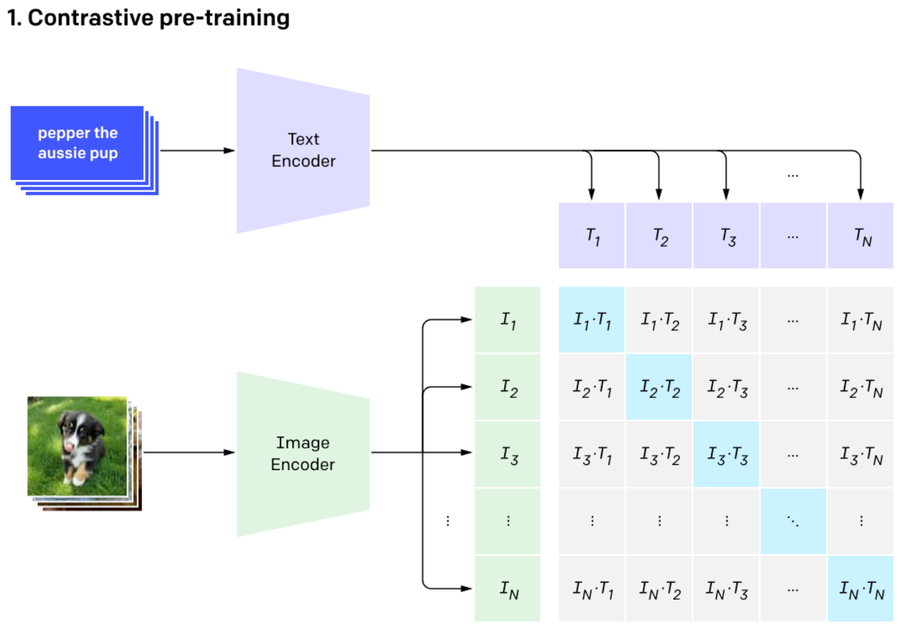
InfoNCE with two modalities: text and image

- Previously, we have seen how InfoNCE was used to learn to classify positive examples where positives were **context from previous time step**, **patches from same image**, **augmented views** from the same image.
- Why not use InfoNCE to “push closer” images and their related captions?

OpenAI’s CLIP model

Contrastive learning applied to image-caption pairs:

- Pre-training step is to train a multi-modal model on matching images and their correct captions.
- No need to manually annotate millions of data (costly)



OpenAI's CLIP model

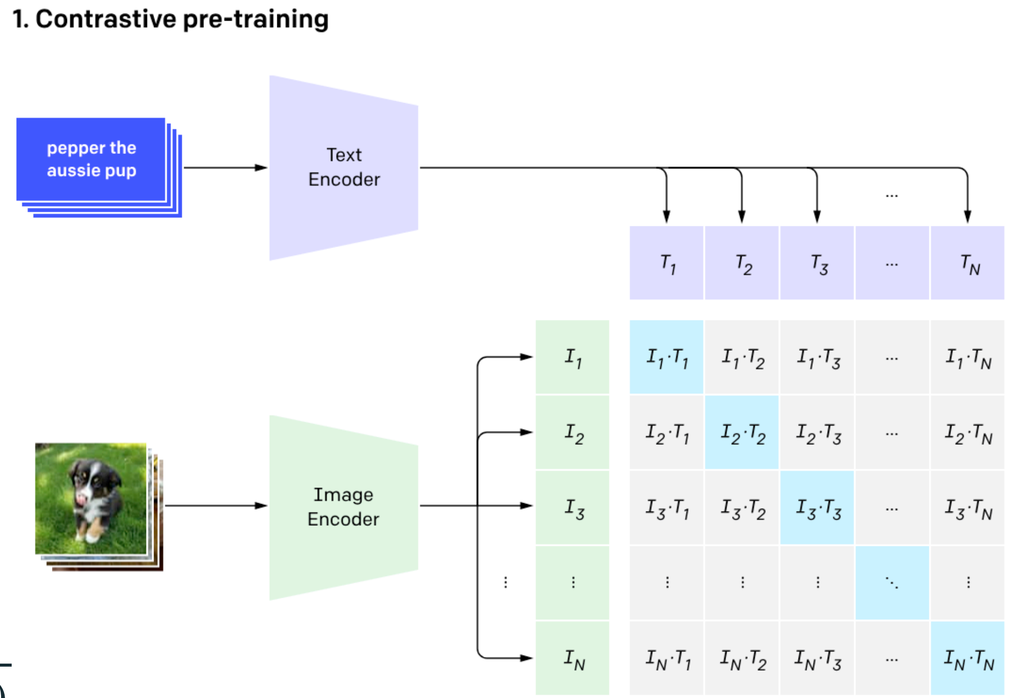
Contrastive learning applied to image-caption pairs:

- Pre-training step is to train a multi-modal model on matching images and their correct captions.
- No need to manually annotate millions of data (costly)

$$\text{InfoNCE} = \frac{1}{2}(\text{InfoNCE}_{img} + \text{InfoNCE}_{text})$$

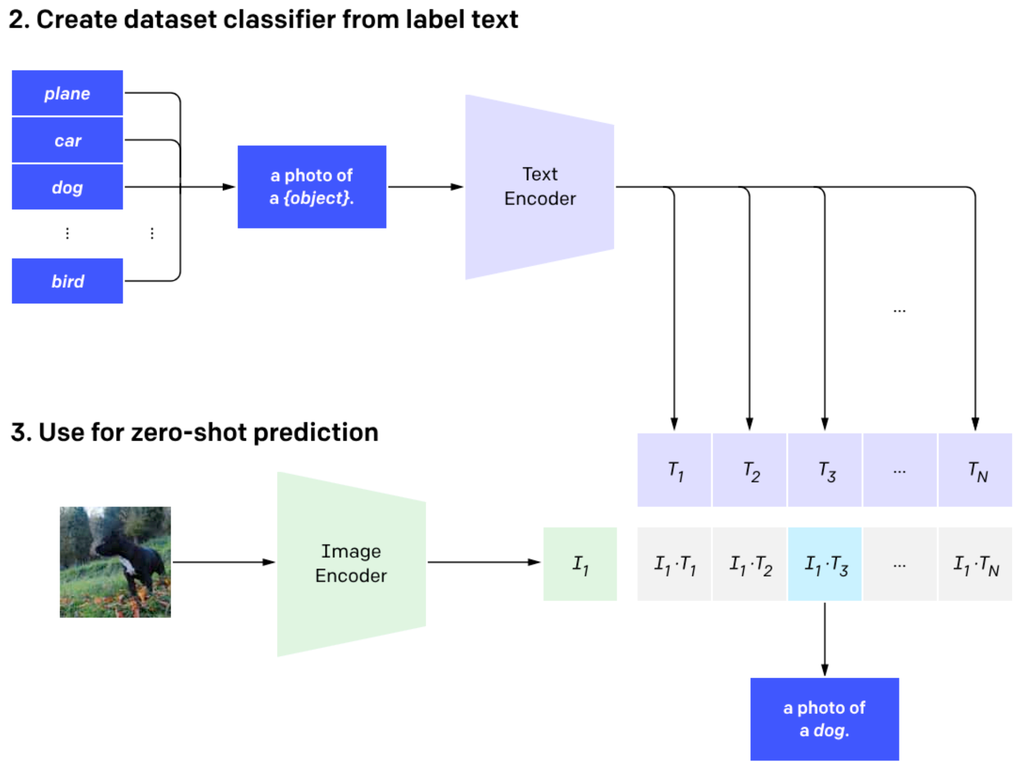
$$= \frac{1}{2} \mathop{\mathbb{E}}_{(x,y)\sim p_{data\ pairs}} \Big[-\log \frac{\exp(f(x)^T g(y)/\tau)}{\exp(f(x)^T g(y))/\tau) + \sum_{y' \neq y} \exp(f(x)^T g(y')/\tau)}$$

$$-\log \frac{\exp(f(x)^T g(y)/\tau)}{\exp(f(x)^T g(y))/\tau) + \sum_{x' \neq x} \exp(f(x')^T g(y)/\tau)} \Big]$$



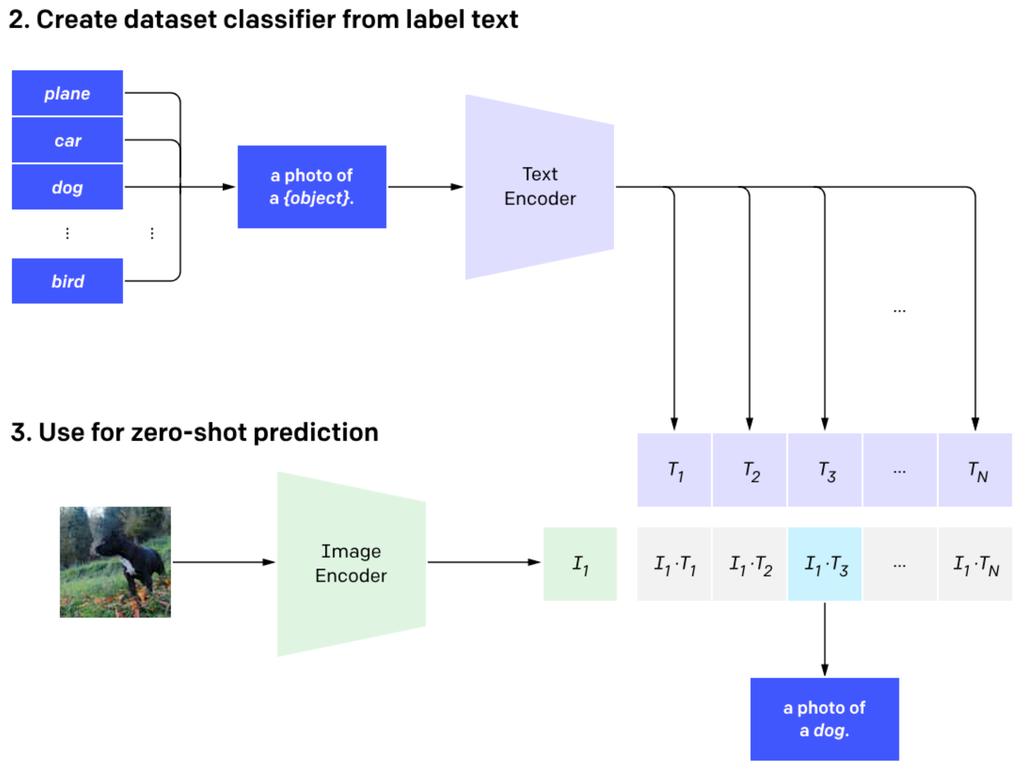
OpenAI’s CLIP model

- Create text classifier by embedding each class names with template(s) and getting per-class text embedding vector.
- For each image, compare its image embedding similarity with each class text's embedding and pick the highest scoring class
- **No linear evaluation or fine-tuning!**



OpenAI's CLIP model

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.



BLIP model: adding a generator

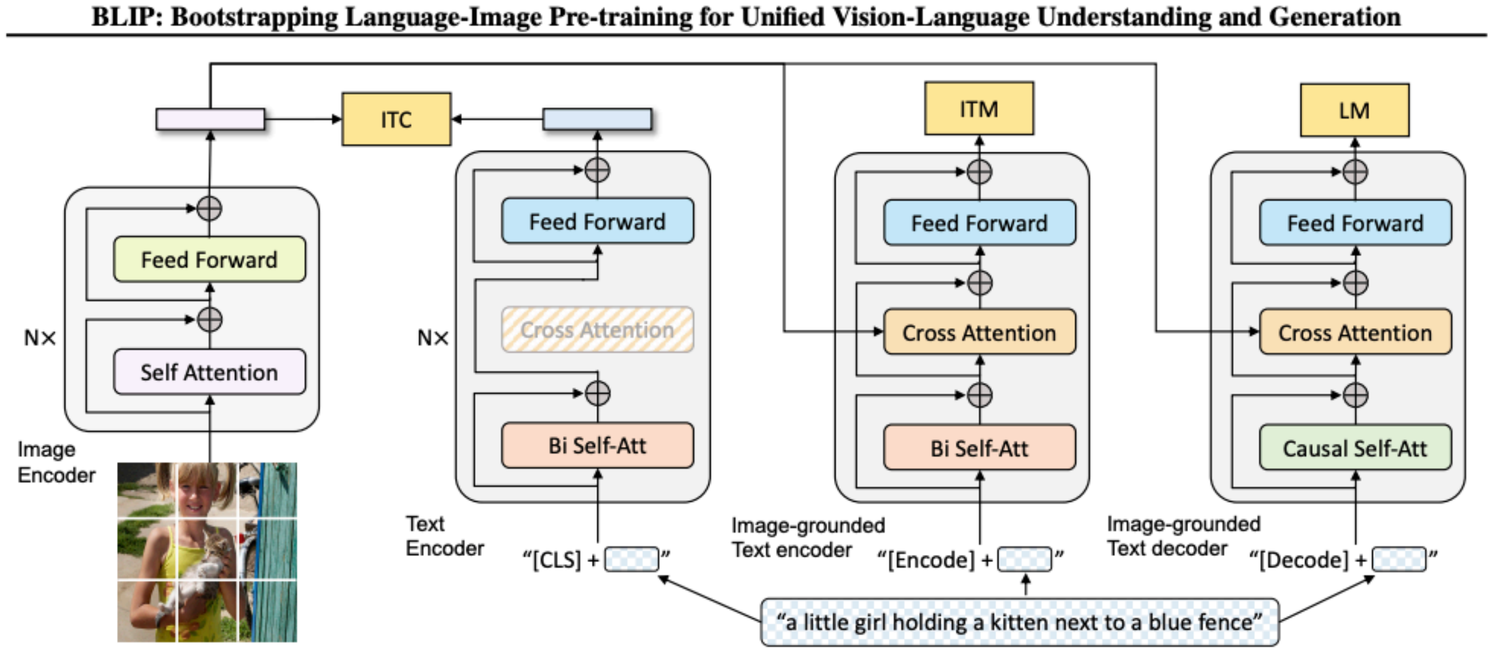


Figure 2. Pre-training model architecture and objectives of BLIP (same parameters have the same color). We propose multimodal mixture of encoder-decoder, a unified vision-language model which can operate in one of the three functionalities: (1) Unimodal encoder is trained with an image-text contrastive (ITC) loss to align the vision and language representations. (2) Image-grounded text encoder uses additional cross-attention layers to model vision-language interactions, and is trained with a image-text matching (ITM) loss to distinguish between positive and negative image-text pairs. (3) Image-grounded text decoder replaces the bi-directional self-attention layers with causal self-attention layers, and shares the same cross-attention layers and feed forward networks as the encoder. The decoder is trained with a language modeling (LM) loss to generate captions given images.

BLIP model: adding a generator

- Contrastive loss (**ITC**) between image and text modalities (same as CLIP)
- Image-Text Matching (**ITM**) with image-grounded text encoder: predict $p([\text{text is relevant}] \mid \text{image})$ i.e. binary version of ITC.
- Language Modeling (**LM**) to train as an image-grounded text decoder: the loss is text generation (cross-entropy) on the correct caption.

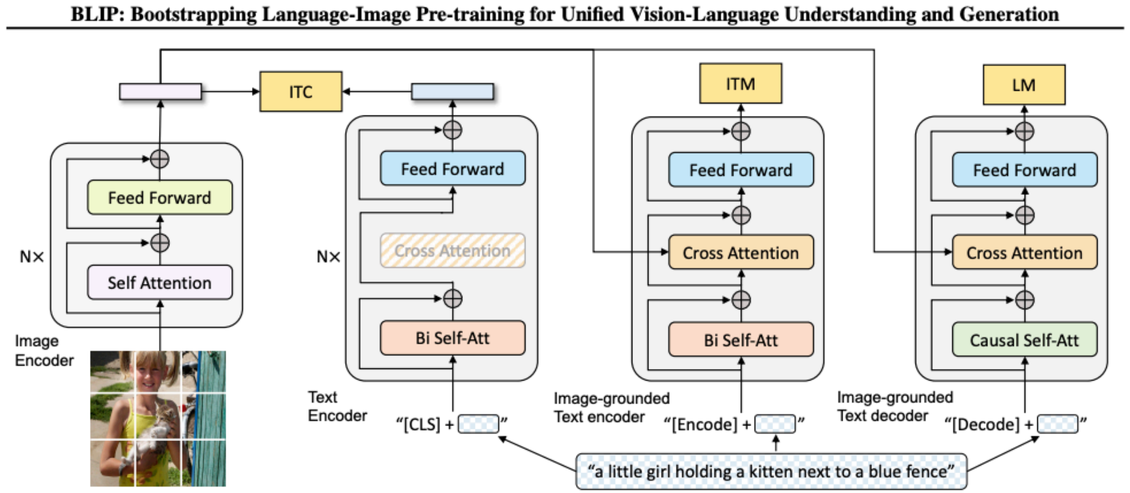


Figure 2. Pre-training model architecture and objectives of BLIP (same parameters have the same color). We propose multimodal mixture of encoder-decoder, a unified vision-language model which can operate in one of the three functionalities: (1) Unimodal encoder is trained with an image-text contrastive (ITC) loss to align the vision and language representations. (2) Image-grounded text encoder uses additional cross-attention layers to model vision-language interactions, and is trained with a image-text matching (ITM) loss to distinguish between positive and negative image-text pairs. (3) Image-grounded text decoder replaces the bi-directional self-attention layers with causal self-attention layers, and shares the same cross-attention layers and feed forward networks as the encoder. The decoder is trained with a language modeling (LM) loss to generate captions given images.

Code assignment

https://colab.research.google.com/drive/1ZUI_qKUvSwanj3Sj_XG4JCBUybMZriQP#scrollTo=v3z3C8UBVGxh

VLMs failures

- E.g. InfoNCE discarding information such as syntax and leading to bag-of-words behaviour of VLMs (see Yuksekgonul et al.)
- Evaluation benchmarks are flawed (see Lin et al.)
- Biases are created by training data leading models to follow text semantics priors (see Lin et al.)

So what's next?

- Contrastive Learning has been and continues to be widely used to train self-supervised architectures.
- However it has some drawbacks, especially for image-text models. In this space, interesting avenues are:
  - Learning objectives for representations that enable *common sense (AGI)*?
  - *Uncertainty estimation*: do models know when they don't know?
  - *Models biases*: how to ensure models perform equally well on disparate subparts of the data (and not just the majority)?

Thank you!

- Questions?
- We’re always looking for students & collaborators!