Self-supervised learning in computer vision and medical imaging

Pietro Gori

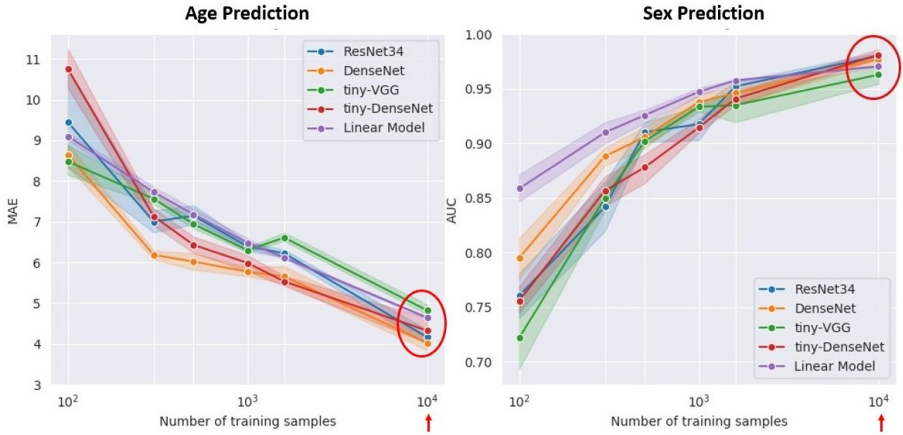
Assistant Professor
Télécom Paris (IPParis)
Paris, France

Summary

Summary

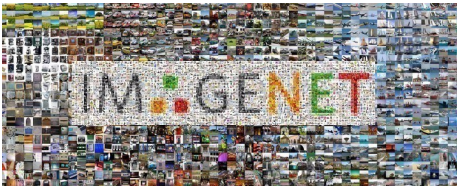
Introduction - Computer Vision

- Deep learning (e.g., CNN or ViT) is a lazy and inefficient statistical method that needs millions if not billions of exemples to learn a precise task → **data hungry**



Introduction - Computer Vision

- Many specific tasks in Computer Vision, such as object detection[1] (e.g., YOLO), image classification[2] (e.g., ResNet-50), or semantic segmentation (e.g., U-Net), have reached astonishing results in the last years.
- This has been possible mainly because **large** ($N > 10^6$), **labeled** data-sets were *easily* accessible and *freely* available



[1] T.-Y. Lin et al. "Microsoft COCO: Common Objects in Context". In: *ECCV*. 2014.

[2] J. Deng et al. "ImageNet: A Large-Scale Hierarchical Image Database". In: *CVPR*. 2009.

Introduction - Medical Imaging

- In medical imaging, current **research** datasets are:
  - **small**: $N < 2k$ for common pathology and $N < 200$ for rare pathology
  - **biased**: images are acquired in a precise hospital, following a specific protocol with a particular machine (nuisance site effect)
  - **multi-modal**: many imaging modalities can be available as well as text, clinical, biological, genetic data.
  - **anonymized**, **quality checked**, **accessible**, **quite homogeneous**

- **Clinical** datasets are harder to analyze since they are usually **not anonymized**, **not quality checked**, **not freely accessible**, **highly heterogeneous**.

- In this talk, we will focus on **research** medical imaging datasets

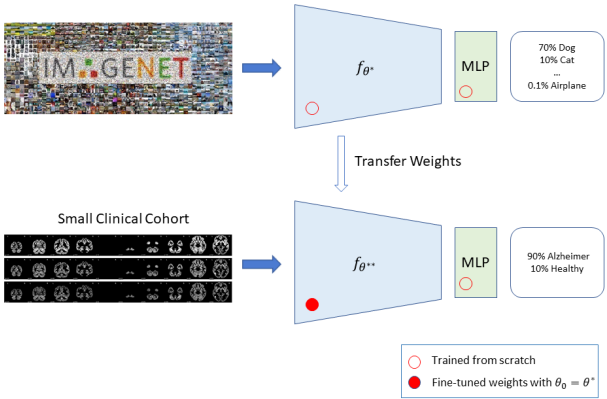
Summary

Introduction - Transfer Learning

- When dealing with small labelled datasets, a common strategy is **Transfer Learning**:
  1. pre-training a model on a *large* dataset and then
  2. fine-tuning it on the *small* target and labelled dataset

Introduction - Transfer Learning

- When dealing with small labelled datasets, a common strategy is **Transfer Learning**:
  1. pre-training a model on a *large* dataset and then
  2. fine-tuning it on the *small* target and labelled dataset
- **Supervised pre-training** from ImageNet is common.



Introduction - Transfer Learning

- When dealing with small labelled datasets, a common strategy is **Transfer Learning**:
  1. pre-training a model on a *large* dataset and then
  2. fine-tuning it on the *small* target and labelled dataset
- **Supervised pre-training** from ImageNet is common. Its usefulness (that is, feature reuse) increases with[3][4][5][6]:
  - reduced target data size (small N_{target})
  - visual similarity between pre-train and target domains (small FID)
  - models with fewer inductive biases (TL works better for ViTs than CNN)
  - larger architectures (more parameters)

[3] B. Mustafa et al. *Supervised Transfer Learning at Scale for Medical Imaging*. 2021.
[4] C. Matsoukas et al. "What Makes Transfer Learning Work for Medical Images". In: *CVPR*. 2022.
[5] B. Neyshabur et al. "What is being transferred in transfer learning?" In: *NeurIPS*. 2020.
[6] M. Raghu et al. "Transfusion: Understanding Transfer Learning for Medical Imaging". In: *NeurIPS*. 2019.

Introduction - Transfer Learning

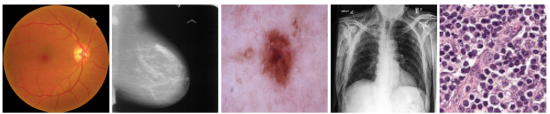
- When dealing with small labelled datasets, a common strategy is **Transfer Learning**:
  1. pre-training a model on a *large* dataset and then
  2. fine-tuning it on the *small* target and labelled dataset

- **Supervised pre-training** from ImageNet is common. Its usefulness (that is, feature reuse) increases with[7][8][9][10]:
  - reduced target data size (small N_{target})
  - visual similarity between pre-train and target domains (small FID)
  - models with fewer inductive biases (TL works better for ViTs than CNN)
  - larger architectures (more parameters)

[7] B. Mustafa et al. *Supervised Transfer Learning at Scale for Medical Imaging*. 2021.
[8] C. Matsoukas et al. "What Makes Transfer Learning Work for Medical Images". In: *CVPR*. 2022.
[9] B. Neyshabur et al. "What is being transferred in transfer learning?" In: *NeurIPS*. 2020.
[10] M. Raghu et al. "Transfusion: Understanding Transfer Learning for Medical Imaging". In: *NeurIPS*. 2019.

Introduction - Transfer Learning

- Natural[11] and Medical[12] images can be visually very different ! → **Domain gap**

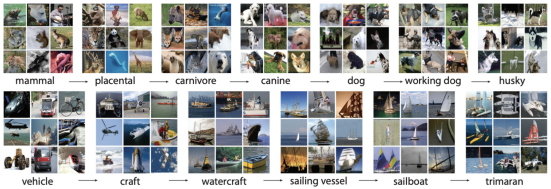


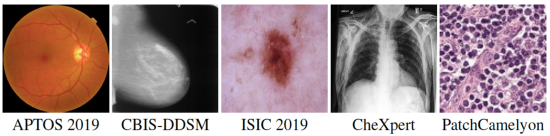
- Furthermore, Medical images can be 3D. ImageNet is 2D.
- Need for **3D**, **annotated**, **large** medical dataset

[11] J. Deng et al. "ImageNet: A Large-Scale Hierarchical Image Database". In: *CVPR*. 2009.
[12] C. Matsoukas et al. "What Makes Transfer Learning Work for Medical Images". In: *CVPR*. 2022.

Introduction - Transfer Learning

- Natural[11] and Medical[12] images can be visually very different ! → **Domain gap**





- Furthermore, Medical images can be 3D. ImageNet is 2D.
- Need for **3D**, **annotated**, **large** medical dataset → PROBLEM !

[11] J. Deng et al. "ImageNet: A Large-Scale Hierarchical Image Database". In: *CVPR*. 2009.
[12] C. Matsoukas et al. "What Makes Transfer Learning Work for Medical Images". In: *CVPR*. 2022.

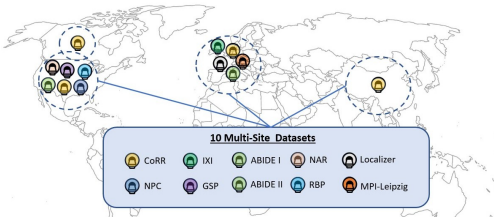
- Supervised pre-training is a not a valid option in medical imaging. Need for another kind of pre-training.

[13] T. J. Littlejohns et al. "The UK Biobank imaging enhancement of 100,000 participants:" in: *Nature Communications* (2020).

[14] B. Dufumier et al. "OpenBHB: a Large-Scale Multi-Site Brain MRI Data-set for Age Prediction and Debiasing". In: *NeuroImage* (2022).

- Supervised pre-training is a not a valid option in medical imaging. Need for another kind of pre-training.
- Recently **big**, **multi-sites** international **healthy data-sets** have emerged, such as UK Biobank[13] ($N > 100k$) and OpenBHB[14] ($N > 10k$)

biobank uk
Enabling scientific discoveries that improve human health

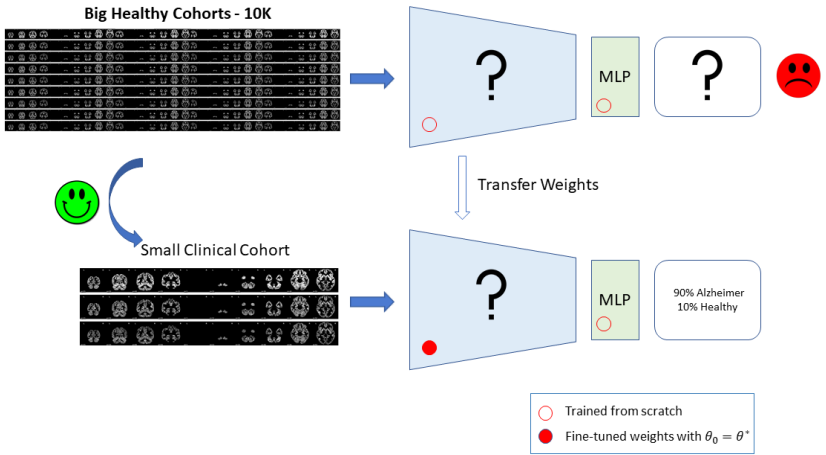


[13] T. J. Littlejohns et al. "The UK Biobank imaging enhancement of 100,000 participants:" in: *Nature Communications* (2020).

[14] B. Dufumier et al. "OpenBHB: a Large-Scale Multi-Site Brain MRI Data-set for Age Prediction and Debiasing". In: *NeuroImage* (2022).

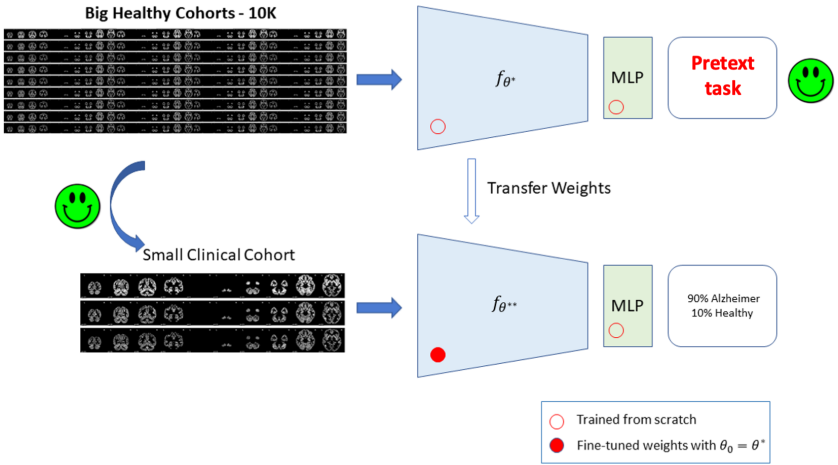
Introduction - Transfer Learning

- How can we employ an **healthy** (thus **unlabeled**) data-set for pre-training ?



Introduction - Transfer Learning

- How can we employ an **healthy** (thus **unlabeled**) data-set for pre-training ? → Self-Supervised pre-training !



- **Self-supervised pre-training**: leverage an annotation-free *pretext task* to provide a surrogate supervision signal for feature learning.

[15] C. Doersch et al. "Unsupervised Visual Representation Learning by Context Prediction". In: *ICCV*. 2015.
[16] K. He et al. "Masked Autoencoders Are Scalable Vision Learners". In: *CVPR*. 2022.
[17] J. Donahue et al. "Large Scale Adversarial Representation Learning". In: *NeurIPS*. 2019.
[18] T. Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations". In: *ICML*. 2020.
[19] J.-B. Grill et al. "Bootstrap your own latent: A new approach to self-supervised Learning". In: *NeurIPS*. 2020.
[20] A. Bardes et al. "VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning". In: *ICLR*. 2022.

Introduction - Transfer Learning

- **Self-supervised pre-training**: leverage an annotation-free *pretext task* to provide a surrogate supervision signal for feature learning.
- Pretext task should only use the visual information and context of the images

[15] C. Doersch et al. "Unsupervised Visual Representation Learning by Context Prediction". In: *ICCV*. 2015.
[16] K. He et al. "Masked Autoencoders Are Scalable Vision Learners". In: *CVPR*. 2022.
[17] J. Donahue et al. "Large Scale Adversarial Representation Learning". In: *NeurIPS*. 2019.
[18] T. Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations". In: *ICML*. 2020.
[19] J.-B. Grill et al. "Bootstrap your own latent: A new approach to self-supervised Learning". In: *NeurIPS*. 2020.
[20] A. Bardes et al. "VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning". In: *ICLR*. 2022.

Introduction - Transfer Learning

- **Self-supervised pre-training**: leverage an annotation-free *pretext task* to provide a surrogate supervision signal for feature learning.
- Pretext task should only use the visual information and context of the images
- Examples of pretext tasks:
  - Context prediction[15]
  - Generative models[16,17]
  - Instance discrimination[18]
  - Teacher/Student[19]
  - Information Maximization[20]

[15] C. Doersch et al. "Unsupervised Visual Representation Learning by Context Prediction". In: *ICCV*. 2015.
[16] K. He et al. "Masked Autoencoders Are Scalable Vision Learners". In: *CVPR*. 2022.
[17] J. Donahue et al. "Large Scale Adversarial Representation Learning". In: *NeurIPS*. 2019.
[18] T. Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations". In: *ICML*. 2020.
[19] J.-B. Grill et al. "Bootstrap your own latent: A new approach to self-supervised Learning". In: *NeurIPS*. 2020.
[20] A. Bardes et al. "VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning". In: *ICLR*. 2022.

Self-supervised Learning - Preliminaries

- Pre-text tasks should produce image representations that are:

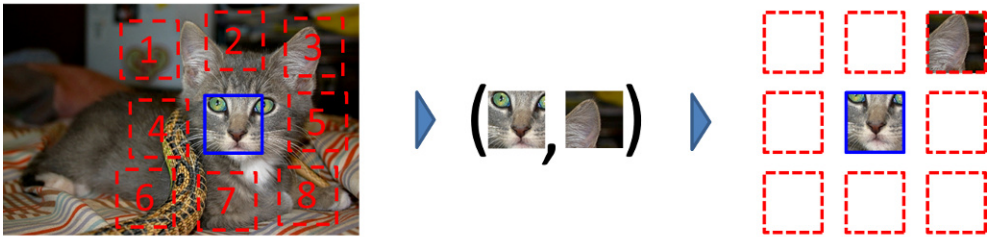
1. **Transferable**: we can easily reuse/fine-tune them in different downstream tasks (e.g., segmentation, object detection, classification, etc.)
2. **Generalizable**: they should not be specific to a single task but work well in several different downstream tasks
3. **High-level**: representations should characterize the high-level semantics/structure and not low-level features (color, texture, etc.)
4. **Invariant**: image representations should be invariant to geometric or appearance transformations that do not modify the information content of the image (i.e., irrelevant for downstream task)
5. **Semantically coherent**: semantically similar images should be close in the representation space

Summary

Summary

Context prediction - Pair of patches

- Given an image, we can divide it into patches and predict their relative position

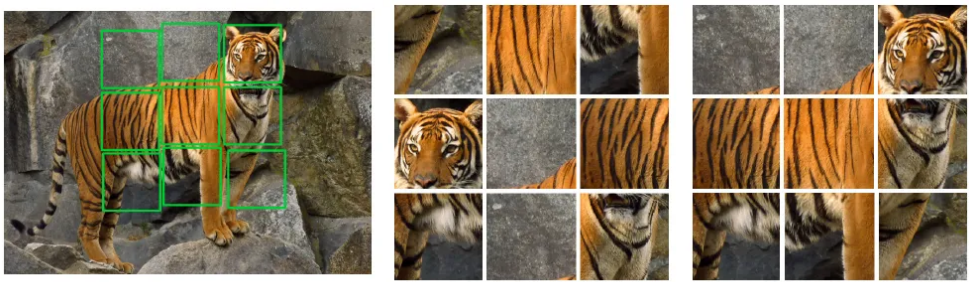


Given a central patch (blue), predict the relative position of the other patch among the 8 possible connected neighbor positions.[21]

- Or we could use all neighbor patches...

[21] C. Doersch et al. "Unsupervised Visual Representation Learning by Context Prediction". In: *ICCV*. 2015.

Context prediction - Jigsaw puzzle



(a) (b) (c)

First randomly shuffle all 9 patches and then learn to solve a Jigsaw puzzle. Considering all patches together remove ambiguities since placement is mutually exclusive. See central patch and top two left patches.[22]

[22] M. Noroozi et al. "Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles". In: *ECCV*. 2016.

Context prediction

- To avoid trivial shortcuts, low-level trivial signals such as boundary patterns or texture continuing between patches:
  - Add gaps between patches
  - Randomly crop each patch
  - Shuffle several times same image (for jigsaw)
  - Shift color channels and/or use grayscale images
  - Add random shifts to each patch
  - Downsample and then upsample images (robustness to pixelation)

Transformation prediction

- Instead than predicting the relative position between two patches of an image, predict the geometric transformation applied to the entire image
- Which transformation? $\rightarrow$ Need for transformations that:
  - force the network (usually CNN) to recognize all objects depicted in the image and learn high-level semantic features (e.g., class, location, size, pose, etc.)
  - do not leave easily detectable low-level visual artifacts when applied to images
  - are well-posed, namely there should not be ambiguity about the original image.

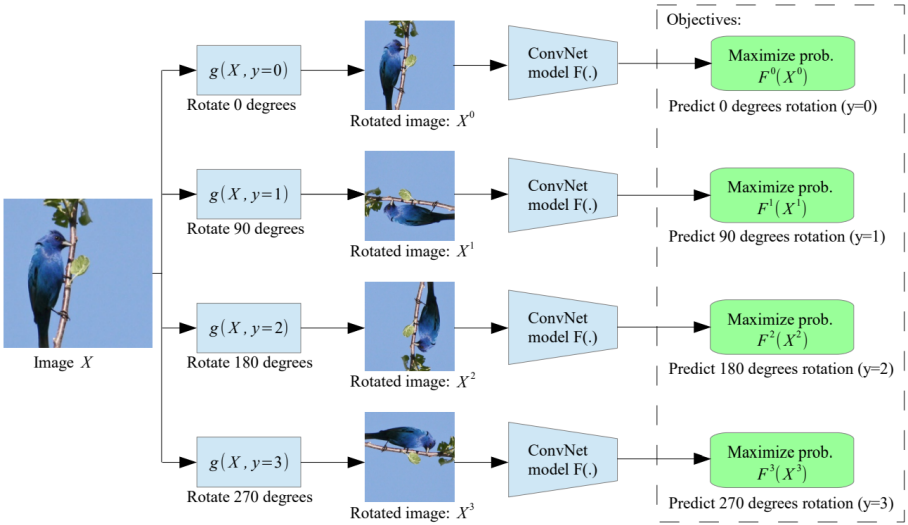
[23] S. Gidaris et al. "Unsupervised Representation Learning by Predicting Image Rotations". In: *ICLR*. 2018.

Transformation prediction

- Instead than predicting the relative position between two patches of an image, predict the geometric transformation applied to the entire image
- Which transformation? $\rightarrow$ Need for transformations that:
  - force the network (usually CNN) to recognize all objects depicted in the image and learn high-level semantic features (e.g., class, location, size, pose, etc.)
  - do not leave easily detectable low-level visual artifacts when applied to images
  - are well-posed, namely there should not be ambiguity about the original image.
- **Proposed solution: Image rotations by 0, 90, 180, 270 degrees**[23]

[23] S. Gidaris et al. "Unsupervised Representation Learning by Predicting Image Rotations". In: *ICLR*. 2018.

Transformation prediction



Transformation prediction

- **No easily detectable low-level visual artifacts**: implementation of rotations only use transpose and flip operations. No interpolations are used which might produce artifacts (note that choice of angles is important !)

Effect of 10 successive rotations of 36 degrees to the original image (left) using Nearest Neighbor interpolation (middle) or Linear interplation (right). Credits: http://bigwww.epfl.ch/demo/jaffine/index.html

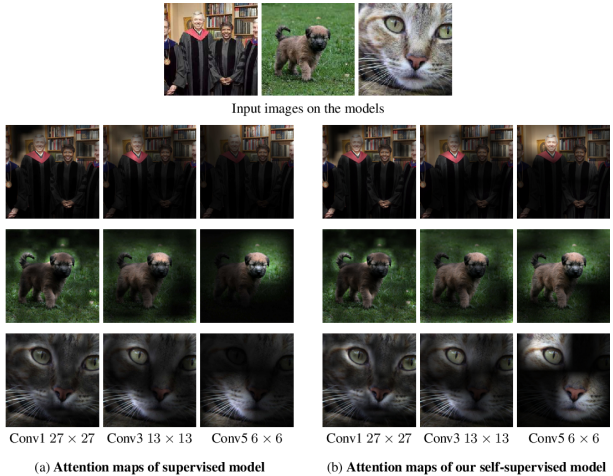
Transformation prediction

- **No easily detectable low-level visual artifacts**: implementation of rotations only use transpose and flip operations. No interpolations are used which might produce artifacts (note that choice of angles is important !)
- **Well-posed transformations**: we assume that all images have a usual and standard position. For instance, when using natural images all objects should be “up-standing”

Transformation prediction

- **No easily detectable low-level visual artifacts**: implementation of rotations only use transpose and flip operations. No interpolations are used which might produce artifacts (note that choice of angles is important !)
- **Well-posed transformations**: we assume that all images have a usual and standard position. For instance, when using natural images all objects should be “up-standing”
- **Detect objects and focus on high-level semantic features**: CNN networks need to focus on high-level features to correctly retrieve the rotation

Transformation prediction



(a) **Attention maps of supervised model** (b) **Attention maps of our self-supervised model**

Attention maps (sum of feature activations raised to the power of 1,2,4 respectively).[24]

[24] S. Gidaris et al. "Unsupervised Representation Learning by Predicting Image Rotations". In: *ICLR*. 2018.

Summary

Generative models

- Generative models use as pretext task the reconstruction of the modified/corrupted/partially observed original image
- The main differences between the methods in the literature are:
  - how you modify/corrupt the original image
  - regularization losses (adversarial loss, sub-networks, etc.)
  - neural network architectural choices (CNN, ViT, etc.)

Here, we will see:

1. Denoising-based autoencoders
2. Colorization-based autoencoders
3. Gan based methods (ALI/ BiGAN / BigBiGAN)

Generative models

- Generative models use as pretext task the reconstruction of the modified/corrupted/partially observed original image
- The main differences between the methods in the literature are:
  - how you modify/corrupt the original image
  - regularization losses (adversarial loss, sub-networks, etc.)
  - neural network architectural choices (CNN, ViT, etc.)

Here, we will see:

1. Denoising-based autoencoders
2. Colorization-based autoencoders
3. Gan based methods (ALI/ BiGAN / BigBiGAN)

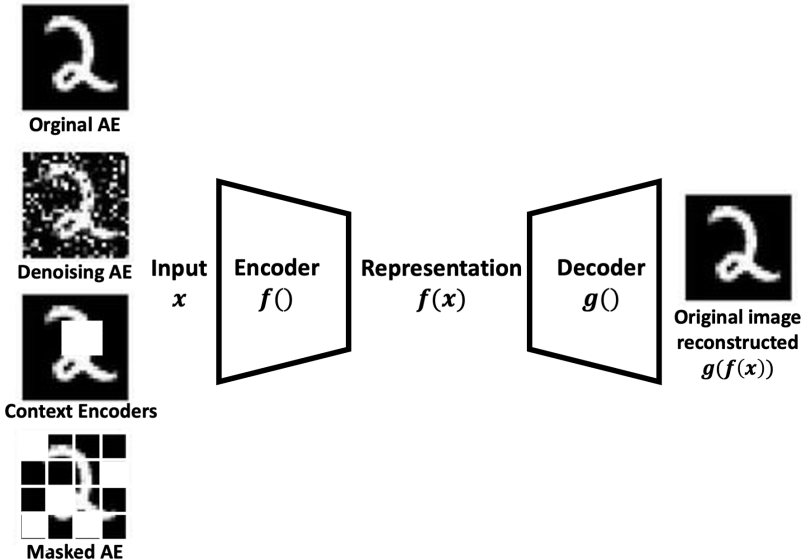
Denoising-based autoencoders

- Usual autoencoders[25] are constituted of an encoder $f()$ and a decoder $g()$ and they are trained to minimize the reconstruction error $||x - g(f(x))||_2^2$, which amounts to maximizing a lower bound on the mutual information between input x and learnt representation $f(x)$[26]
- To avoid trivial solutions (i.e., $g(f(\cdot)) = I$ identity mapping), the representation $f(x)$ is usually of lower dimension than x (bottleneck), thus compressing the information
- To extract more useful features (i.e., representations), different methods were proposed to reconstruct a "repaired" version of a corrupted original image. The key question is, **how do we corrupt/modify the original image ?**

[25] G. E. Hinton et al. "Reducing the Dimensionality of Data with Neural Networks". In: *Science* (2006).

[26] P. Vincent et al. "Stacked Denoising Autoencoders: Learning Useful Representations ...". In: *JMLR* (2010).

Denoising-based autoencoders



Denoising-based autoencoders

- **Denoising autoencoders**[27]: learn to restore a corrupted image where Gaussian/salt-and-pepper noise (or others) has been applied.

- **Context Encoder/ Inpainting**[28]: learn to regress pixel values of a large, single missing region using CNN, L2 reconstruction and adversarial losses.

- **Masked image modeling**[29]: randomly mask some patches of the image and predict the masked patches from the visible ones using Transformers (e.g., ViT).

[27] P. Vincent et al. "Extracting and composing robust features with denoising autoencoders". In: *ICML*. 2008.
[28] D. Pathak et al. "Context Encoders: Feature Learning by Inpainting". In: *CVPR*. 2016.
[29] K. He et al. "Masked Autoencoders Are Scalable Vision Learners". In: *CVPR*. 2022.

Denoising-based autoencoders

- **Denoising autoencoders**[27]: learn to restore a corrupted image where Gaussian/salt-and-pepper noise (or others) has been applied.
  Problem: localized and low-level corruption, no semantic information needed to repair image
- **Context Encoder/ Inpainting**[28]: learn to regress pixel values of a large, single missing region using CNN, L2 reconstruction and adversarial losses.
  Problem: 1) image synthesis quality is difficult to evaluate, 2) domain gap between training set (images with holes) and test set (full images), 3) if the mask is not big, no high-level reasoning, just copy low and mid-level neighboring structures
- **Masked image modeling**[29]: randomly mask some patches of the image and predict the masked patches from the visible ones using Transformers (e.g., ViT).

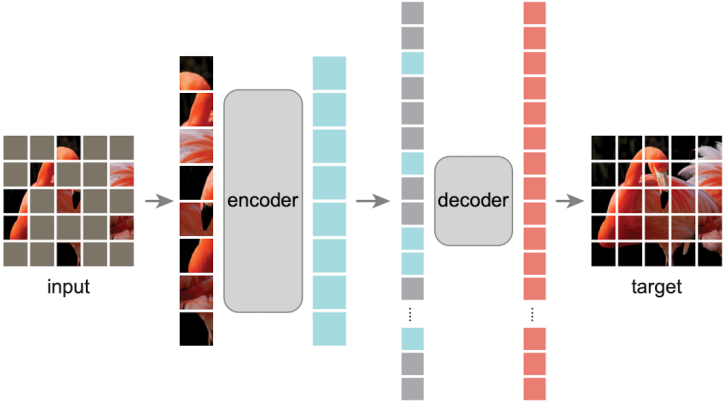
[27] P. Vincent et al. "Extracting and composing robust features with denoising autoencoders". In: *ICML*. 2008.
[28] D. Pathak et al. "Context Encoders: Feature Learning by Inpainting". In: *CVPR*. 2016.
[29] K. He et al. "Masked Autoencoders Are Scalable Vision Learners". In: *CVPR*. 2022.

Masked image modeling

- Many methods took inspiration from **Causal language modeling**, such as GPT[30], where masked tokens are predicted using only previous (left) tokens, and **Masked language modeling**, such as **BERT**[31], where masked tokens are recovered using both previous (left) and past (right) tokens
- A direct translation of GPT and BERT from language to vision works well but 1) it is computationally too demanding and 2) it under-performs wrt other contrastive methods[32]
- Recent works (BEiT[33], SimMIM[34] and **MAE**[35]) found that the use of ViT, great masking portion and lightweight decoder resulted in SOTA performance.

[30] A. Radford et al. *Language Models are Unsupervised Multitask Learners.*
[31] J. Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.*
[32] M. Chen et al. "Generative Pretraining From Pixels". In: *ICML*. 2020.
[33] H. Bao et al. "BEiT: BERT Pre-Training of Image Transformers". In: *ICLR*. 2022.
[34] Z. Xie et al. "SimMIM: a Simple Framework for Masked Image Modeling". In: *CVPR*. 2022.
[35] K. He et al. "Masked Autoencoders Are Scalable Vision Learners". In: *CVPR*. 2022.

Masked Autoencoders



Masked Autoencoders[36]: encoder ViT takes as input only visible patches. Lightweight decoder takes as input the encoded visible patches and the masked patches (as shared, learned, constant representations) with their positional embeddings.

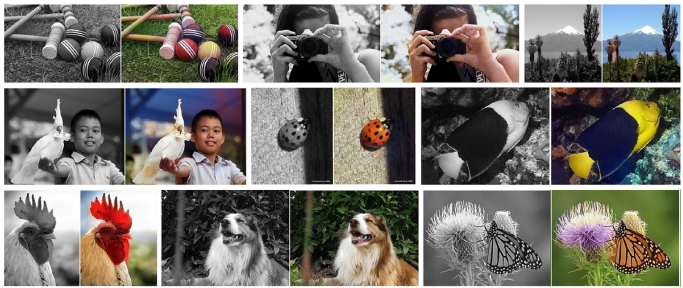
[36] K. He et al. "Masked Autoencoders Are Scalable Vision Learners". In: *CVPR*. 2022.

Generative models

1. Denoising-based autoencoders
2. Colorization-based autoencoders
3. Gan based methods (ALI/ BiGAN / BigBiGAN)

Colorization-based autoencoders

- Another way to extract useful features in autoencoders is through **colorization**: the encoder takes as input only one channel of the original image (e.g., grayscale/intensity) and the decoder needs to predict the original colors (e.g., RGB/Lab channels)



Choices of training set, color channels, architecture and optimization are important.[37,38,39]

[37] R. Zhang et al. "Colorful Image Colorization". In: *ECCV*. 2016.
[38] G. Larsson et al. "Learning Representations for Automatic Colorization". In: *ECCV*. 2016.
[39] G. Larsson et al. "Colorization as a Proxy Task for Visual Understanding". In: *CVPR*. 2017.

Colorization-based autoencoders

Pros

- systematic, and not stochastic, corruption of image remove the pre-training and testing domain gap.
- Denoising/Inpainting/Masking may use only textural/positional information. Predicting color requires object-level reasoning and thus higher semantic representations.

Cons

- color does not always carry important semantic information, as in medical imaging
- Different channels of the input data are not treated equally

Colorization-based autoencoders

Pros

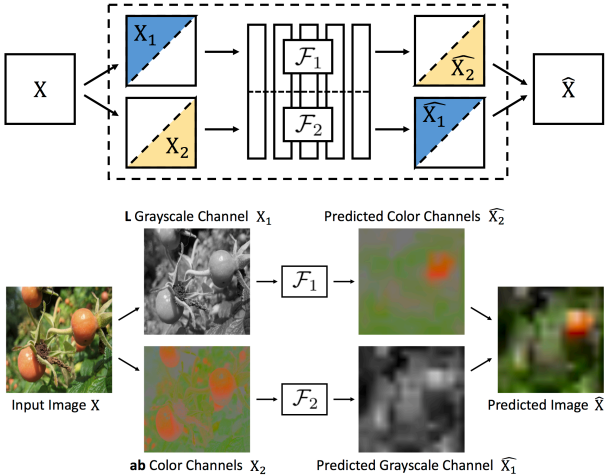
- systematic, and not stochastic, corruption of image remove the pre-training and testing domain gap.
- Denoising/Inpainting/Masking may use only textural/positional information. Predicting color requires object-level reasoning and thus higher semantic representations.

Cons

- color does not always carry important semantic information, as in medical imaging
- Different channels of the input data are not treated equally → Possible architectural solution: use multiple sub-networks trained on different but *complementary* channels[a]

[a] R. Zhang et al. "Split-Brain Autoencoders: Unsupervised Learning by Cross-Channel Prediction". In: *CVPR*. 2017.

Colorization-based autoencoders



Use L channel to predict ab channel and viceversa. Concatenate $\hat{X}_1$ and $\hat{X}_2$ to obtain $\hat{X}$[40].

[40] R. Zhang et al. "Split-Brain Autoencoders: Unsupervised Learning by Cross-Channel Prediction". In: *CVPR*. 2017.

Generative models

1. Denoising-based autoencoders
2. Colorization-based autoencoders
3. Gan based methods (ALI/ BiGAN / BigBiGAN)

Gan based methods

- Autoencoders (AE) and Variational Autoencoders (VAE)[41] are easy to train but tend to produce blurry images
- Other generative models have emerged such as Generative Adversarial Network (GAN)[42] where a *generator* model G learns a mapping from random variables to input data and a *discriminator* model D learns to distinguish the real data from the fake ones produced by G. The models are trained in a minimax game where G tries to fool D.
- GAN models are known for potentially unstable training and less diversity in generation due to their adversarial training nature. For this reason, other models have been proposed such as Normalizing flows[43] and Diffusion models[44].

[41] D. P. Kingma et al. "Auto-Encoding Variational Bayes". In: *ICLR*. 2014.
[42] I. Goodfellow et al. "Generative Adversarial Nets". In: *NIPS*. 2014.
[43] D. J. Rezende et al. "Variational Inference with Normalizing Flows". In: *ICML*. 2015.
[44] J. Ho et al. "Denoising Diffusion Probabilistic Models". In: *NeurIPS*. 2020.

Gan based methods

- However, GANs can not be used to learn rich feature representations in an unsupervised way since they "just" generate images and do not have a bottleneck representation, as in AE/VAE → How can we use GAN to learn a representation?

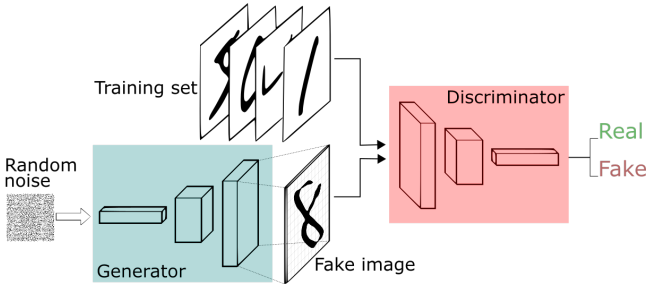
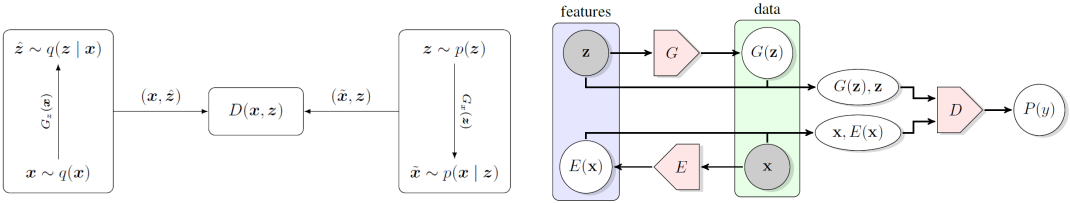


Image from[45].

[45] S. Thalles Santos. *A Short Introduction to Generative Adversarial Networks.* URL: https://sthalles.github.io.

Gan based methods

- However, GANs can not be used to learn rich feature representations in an unsupervised way since they "just" generate images and do not have a bottleneck representation, as in AE/VAE → How can we use GAN to learn a representation?
- Two methods, Adversarially Learned Inference (ALI)[45] and bidirectional GAN (BiGAN)[46], concurrently proposed to add an encoder E which maps real data to latent representations, the inverse of the mapping learned by the generator G



ALI on the left and BiGAN on the right.

[45] V. Dumoulin et al. "Adversarially Learned Inference". In: *ICLR*. 2017.

[46] J. Donahue et al. "Adversarial Feature Learning". In: *ICLR*. 2017.

Algorithm 1 The ALI training procedure.

$\theta_g, \theta_d \leftarrow$ initialize network parameters

repeat

$\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(M)} \sim q(\boldsymbol{x})$ ▷ Draw M samples from the dataset and the prior

$\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(M)} \sim p(\boldsymbol{z})$

$\hat{\boldsymbol{z}}^{(i)} \sim q(\boldsymbol{z} \mid \boldsymbol{x} = \boldsymbol{x}^{(i)}), \quad i = 1, \ldots, M$ ▷ Sample from the conditionals

$\tilde{\boldsymbol{x}}^{(j)} \sim p(\boldsymbol{x} \mid \boldsymbol{z} = \boldsymbol{z}^{(j)}), \quad j = 1, \ldots, M$

$\rho_q^{(i)} \leftarrow D(\boldsymbol{x}^{(i)}, \hat{\boldsymbol{z}}^{(i)}), \quad i = 1, \ldots, M$ ▷ Compute discriminator predictions

$\rho_p^{(j)} \leftarrow D(\tilde{\boldsymbol{x}}^{(j)}, \boldsymbol{z}^{(j)}), \quad j = 1, \ldots, M$

$\mathcal{L}_d \leftarrow -\frac{1}{M} \sum_{i=1}^{M} \log(\rho_q^{(i)}) - \frac{1}{M} \sum_{j=1}^{M} log(1 - \rho_p^{(j)})$ ▷ Compute discriminator loss

$\mathcal{L}_g \leftarrow -\frac{1}{M} \sum_{i=1}^{M} \log(1 - \rho_q^{(i)}) - \frac{1}{M} \sum_{j=1}^{M} \log(\rho_p^{(j)})$ ▷ Compute generator loss

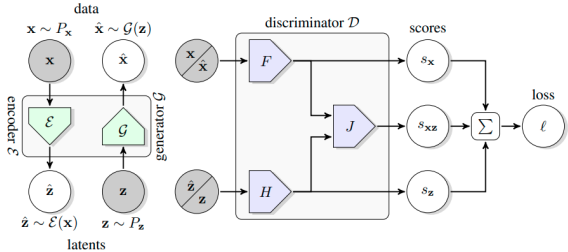
$\theta_d \leftarrow \theta_d - \nabla_{\theta_d} \mathcal{L}_d$ ▷ Gradient update on discriminator network

$\theta_g \leftarrow \theta_g - \nabla_{\theta_g} \mathcal{L}_g$ ▷ Gradient update on generator networks

until convergence

Gan based methods

- BiGAN used a simple generator (DCGAN[47]) which can't produce high-quality images and thus can not capture all visual information → poor representation
- BigBiGAN[48] uses BigGAN[49] as generator producing SOTA representation results

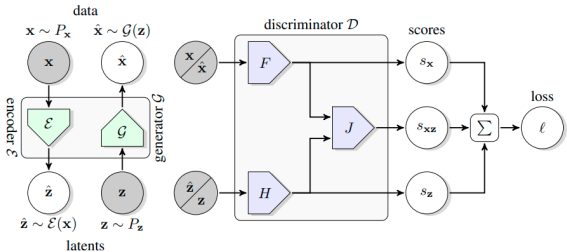


D has three submodules: two unary F(x), H(z) and a joint one J(F(x),H(z)).

[47] A. Radford et al. "Unsupervised Representation Learning with Deep Convolutional GAN". In: *ICLR*. 2016.
[48] J. Donahue et al. "Large Scale Adversarial Representation Learning". In: *NeurIPS*. 2019.
[49] A. Brock et al. "Large Scale GAN Training for High Fidelity Natural Image Synthesis". In: *ICLR*. 2019.

Gan based methods

- BiGAN used a simple generator (DCGAN[47]) which can't produce high-quality images and thus can not capture all visual information → poor representation
- BigBiGAN[48] uses BigGAN[49] as generator producing SOTA representation results → More powerful generator can improve the representation quality



D has three submodules: two unary F(x), H(z) and a joint one J(F(x),H(z)).

[47] A. Radford et al. "Unsupervised Representation Learning with Deep Convolutional GAN". In: *ICLR*. 2016.
[48] J. Donahue et al. "Large Scale Adversarial Representation Learning". In: *NeurIPS*. 2019.
[49] A. Brock et al. "Large Scale GAN Training for High Fidelity Natural Image Synthesis". In: *ICLR*. 2019.

Summary

Instance discrimination

- Usual supervised classification models look for discriminative features that characterize and correctly separate *classes* of objects
- Can we learn a representation where we correctly separate *single instances* (i.e., images) and not classes ?

We'll see four methods:

1. Exemplar-CNN[50]
2. Learning with a non-parametric classifier from a memory bank[51]
3. Learning to count[52]
4. Contrastive Learning[53]

[50] A. Dosovitskiy et al. "Discriminative Unsupervised Feature Learning with Exemplar CNNs". In: *IEEE TPAMI* (2016).
[51] Z. Wu et al. "Unsupervised Feature Learning via Non-parametric Instance Discrimination". In: *CVPR*. 2018.
[52] M. Noroozi et al. "Representation Learning by Learning to Count". In: *ICCV*. 2017.
[53] T. Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations". In: *ICML*. 2020.

Instance discrimination

- Usual supervised classification models look for discriminative features that characterize and correctly separate *classes* of objects
- Can we learn a representation where we correctly separate *single instances* (i.e., images) and not classes ?

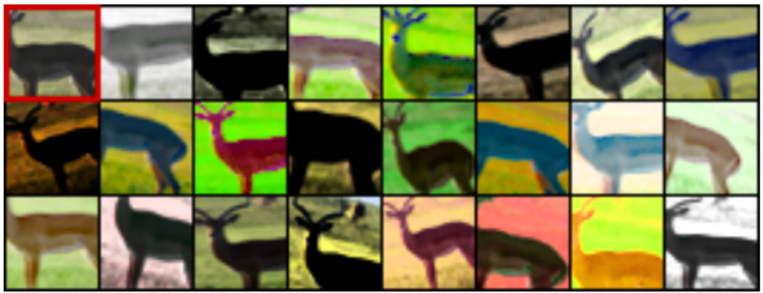
We'll see four methods:

1. Exemplar-CNN[50]
2. Learning with a non-parametric classifier from a memory bank[51]
3. Learning to count[52]
4. Contrastive Learning[53]

[50] A. Dosovitskiy et al. "Discriminative Unsupervised Feature Learning with Exemplar CNNs". In: *IEEE TPAMI* (2016).
[51] Z. Wu et al. "Unsupervised Feature Learning via Non-parametric Instance Discrimination". In: *CVPR*. 2018.
[52] M. Noroozi et al. "Representation Learning by Learning to Count". In: *ICCV*. 2017.
[53] T. Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations". In: *ICML*. 2020.

Exemplar-CNN

- Given a set of different images (or patches) x_i, we first randomly transform K times each image x_i producing a set of transformed images $S_{x_i}{}^{54}$



Exemplar-CNN

- Given a set of different images (or patches) x_i, we first randomly transform K times each image x_i producing a set of transformed images S_{x_i} [54]
- Each set S_{x_i} is then considered as a surrogate class with label i and we minimize the classification loss:

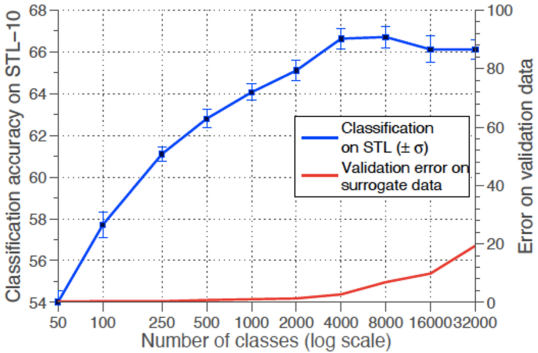
$$\mathcal{L}(X) = \sum_i \sum_k \mathbf{l}(i, T_k x_i) \tag{1}$$

- where T_k is the k-th transformation applied to image x_i and $\mathbf{l}(i, T_k x_i)$ is the cross-entropy (or negative log-likelihood).

[54] A. Dosovitskiy et al. "Discriminative Unsupervised Feature Learning with Exemplar CNNs". In: *IEEE TPAMI* (2016).

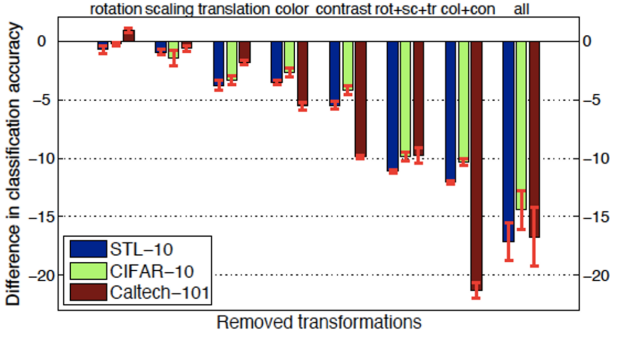
Exemplar-CNN

- The classification loss of Exemplar-CNN ensures that *different images can be distinguished* and it enforces *invariance to (specified) transformations*
- **Number of training data**: classification accuracy increases until an optimum after which it is likely to draw very similar training images that can be hard to discriminate



Exemplar-CNN

- The classification loss of Exemplar-CNN ensures that *different images can be distinguished* and it enforces *invariance to (specified) transformations*
- **Number of training data**: classification accuracy increases until an optimum after which it is likely to draw very similar training images that can be hard to discriminate
- **Influence of Transformations**: classification accuracy varies depending on the used transformations and on the data-set



Instance discrimination

1. Exemplar-CNN[55]
2. Learning with a non-parametric classifier from a memory bank[56]
3. Learning to count[57]
4. Contrastive Learning[58]

[55] A. Dosovitskiy et al. “Discriminative Unsupervised Feature Learning with Exemplar CNNs”. In: *IEEE TPAMI* (2016).
[56] Z. Wu et al. “Unsupervised Feature Learning via Non-parametric Instance Discrimination”. In: *CVPR*. 2018.
[57] M. Noroozi et al. “Representation Learning by Learning to Count”. In: *ICCV*. 2017.
[58] T. Chen et al. “A Simple Framework for Contrastive Learning of Visual Representations”. In: *ICML*. 2020.

Parametric Vs non-parametric classifier

- In Exemplar-CNN we use a parametric classifier based on a softmax.
- Calling W the weights of the last linear layer and $f(x)$ the input to the last linear layer (i.e., representation features of image x), the output of the last linear layer (before softmax) is their matrix product $Wf(x)$
- The loss of Exemplar-CNN is the cross-entropy, namely $\mathbf{l}(i, x) = -\langle y_i, log(g(x))\rangle$, where y_i is the one-hot binary vector ($[0, 0, 1, ..., 0]$ where 1 is at position i) and $g(x) = \text{softmax}(Wf(x))$
- Each row w_i of the matrix W is a classifier of class (here training sample) i. It can be seen as a template/prototype of class i. The softmax function $g(x)$ gives the probability of image x to be the i-th training sample:

$$P(i|f(x); W) = \frac{\exp(w_i^T f(x))}{\sum_j \exp(w_j^T f(x))} \tag{2}$$

Parametric Vs non-parametric classifier

$$P(i|f(x); W) = \frac{\exp(w_i^T f(x))}{\sum_j \exp(w_j^T f(x))} \tag{3}$$

- is a parametric classifier since it depends on W. Can we remove it ?[59]

$$P(i|f(x)) = \frac{\exp(f(x_i)^T f(x)/\tau)}{\sum_j \exp(f(x_j)^T f(x)/\tau)} \quad \text{s.t.} \quad ||f(x_t)||_2 = 1 \quad \forall t \tag{4}$$

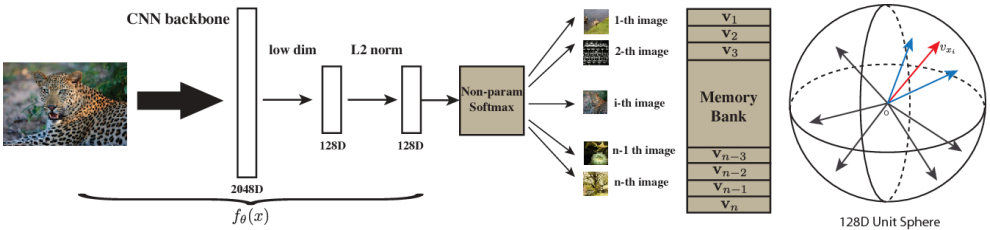
- It thus become a non-parametric classifier where:
  - it does not depend on $W \rightarrow$ no need to compute and store gradient wrt W
  - we directly compare pair of instances instead than template/prototype
  - the temperature τ controls the concentration of instances on the unit hyper-sphere $\rightarrow$ important hyper-parameter

[59] Z. Wu et al. "Unsupervised Feature Learning via Non-parametric Instance Discrimination". In: *CVPR*. 2018.

Non-parametric classifier

- The loss function is defined as the negative log-likelihood over the training set:

$$\mathcal{L}(X) = -\sum_i \log P(i|f(x_i)) \tag{5}$$



The optimal feature embedding is learned via instance-level discrimination, which tries to maximally scatter the features of training samples over the unit sphere. A memory bank is used to limit computations. At each iteration, only the representation $f(x_i)$ and the network parameters are optimized. The representations $f(x_j)$ of the other samples are kept fixed.[60]

[60] S. Gidaris et al. "Unsupervised Representation Learning by Predicting Image Rotations". In: *ICLR*. 2018.

Instance discrimination

1. Exemplar-CNN[61]
2. Learning with a non-parametric classifier from a memory bank[62]
3. Learning to count[63]
4. Contrastive Learning[64]

[61] A. Dosovitskiy et al. "Discriminative Unsupervised Feature Learning with Exemplar CNNs". In: *IEEE TPAMI* (2016).
[62] Z. Wu et al. "Unsupervised Feature Learning via Non-parametric Instance Discrimination". In: *CVPR*. 2018.
[63] M. Noroozi et al. "Representation Learning by Learning to Count". In: *ICCV*. 2017.
[64] T. Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations". In: *ICML*. 2020.

Learning to count

- An image contains a certain number of *visual primitives* that describe and characterize its information content $\rightarrow$ their number should not vary when applying geometric transformations (scaling, rotation, translation, etc.)

[65] M. Noroozi et al. "Representation Learning by Learning to Count". In: *ICCV*. 2017.

Learning to count

- An image contains a certain number of *visual primitives* that describe and characterize its information content $\rightarrow$ their number should not vary when applying geometric transformations (scaling, rotation, translation, etc.)
- Instead, when dividing the image into half or more parts, the number of visual primitives should decrease

[65] M. Noroozi et al. "Representation Learning by Learning to Count". In: *ICCV*. 2017.

Learning to count

- An image contains a certain number of *visual primitives* that describe and characterize its information content → their number should not vary when applying geometric transformations (scaling, rotation, translation, etc.)
- Instead, when dividing the image into half or more parts, the number of visual primitives should decrease
- We look for a representation (i.e., model) f that correctly counts the number of visual primitives of an image

[65] M. Noroozi et al. "Representation Learning by Learning to Count". In: *ICCV*. 2017.

Learning to count

- An image contains a certain number of *visual primitives* that describe and characterize its information content $\rightarrow$ their number should not vary when applying geometric transformations (scaling, rotation, translation, etc.)
- Instead, when dividing the image into half or more parts, the number of visual primitives should decrease
- We look for a representation (i.e., model) f that correctly counts the number of visual primitives of an image
- If we consider as transformations the down-sampling operator $\mathcal{D}$ and the tiling (regular crop) operator $\mathcal{T}_j$, we look for a representation f where[65]

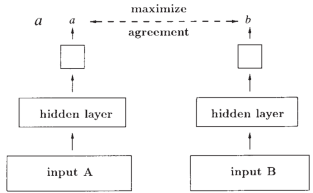
$$f(\mathcal{D} \circ x) = \sum_j f(\mathcal{T}_j \circ x) \tag{6}$$

[65] M. Noroozi et al. "Representation Learning by Learning to Count". In: *ICCV*. 2017.

Learning to count

- In[66] authors propose to use a **Siamese architecture**, initially proposed in 1992/1993 in[67,68], where the same network f maximizes agreement between $f(\mathcal{D} \circ x)$ and $\sum_j f(\mathcal{T}_j \circ x)$ by using as loss $\mathcal{L}(X) = ||f(\mathcal{D} \circ x) - \sum_j f(\mathcal{T}_j \circ x)||_2^2$.

Do you see any problem ? Trivial solution?



[66] M. Noroozi et al. "Representation Learning by Learning to Count". In: *ICCV*. 2017.

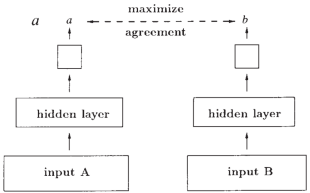
[67] S. Becker et al. "Self-organizing neural network that discovers surfaces in random ...". In: *Nature* (1992).

[68] J. Bromley et al. "Signature Verification using a "Siamese" Time Delay Neural Network". In: *NIPS*. vol. 6. 1993.

Learning to count

- In[66] authors propose to use a **Siamese architecture**, initially proposed in 1992/1993 in[67,68], where the same network f maximizes agreement between $f(\mathcal{D} \circ x)$ and $\sum_j f(\mathcal{T}_j \circ x)$ by using as loss $\mathcal{L}(X) = ||f(\mathcal{D} \circ x) - \sum_j f(\mathcal{T}_j \circ x)||_2^2$.

Do you see any problem ? Trivial solution?



- It may produce the trivial solution $f(x) = 0 \; \forall \, x$, which results in the global minimum 0

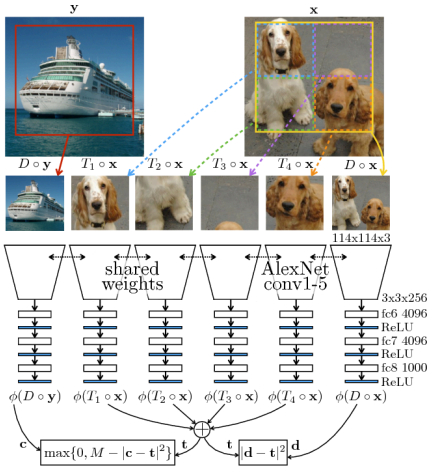
[66] M. Noroozi et al. "Representation Learning by Learning to Count". In: *ICCV*. 2017.

[67] S. Becker et al. "Self-organizing neural network that discovers surfaces in random ...". In: *Nature* (1992).

[68] J. Bromley et al. "Signature Verification using a "Siamese" Time Delay Neural Network". In: *NIPS*. vol. 6. 1993.

Learning to count

- To this end, a possible solution is using a **pairwise contrastive loss** where we enforce that the number of visual primitives should be different between two different images *x* and *y*:



Learning to count

- In the **pairwise contrastive loss**, we force:
  1. transformations of x, that should preserve the number of visual features, to be mapped to the same point in the representation space
  2. transformations of different samples, x and y, to have a distance greater than a positive margin ϵ ($\epsilon \geq 0$) in the representation space: $||f(\mathcal{D} \circ y) - \sum_j f(\mathcal{T}_j \circ x)||_2^2 > \epsilon$

$$\arg\min_f \mathcal{L}(x, y) = \begin{cases} ||f(\mathcal{D} \circ x) - \sum_j f(\mathcal{T}_j \circ x)||_2^2 \\ \max(0, \epsilon - ||f(\mathcal{D} \circ y) - \sum_j f(\mathcal{T}_j \circ x)||_2^2 \end{cases} \tag{7}$$

- **Problems:**
  - Why just using the down-sampling operator $\mathcal{D}$ and the tiling operator $\mathcal{T}_j$?
  - Why using a single "negative" sample y ?

Learning to count

- In the **pairwise contrastive loss**, we force:
  1. transformations of x, that should preserve the number of visual features, to be mapped to the same point in the representation space
  2. transformations of different samples, x and y, to have a distance greater than a positive margin ϵ ($\epsilon \geq 0$) in the representation space: $||f(\mathcal{D} \circ y) - \sum_j f(\mathcal{T}_j \circ x)||_2^2 > \epsilon$

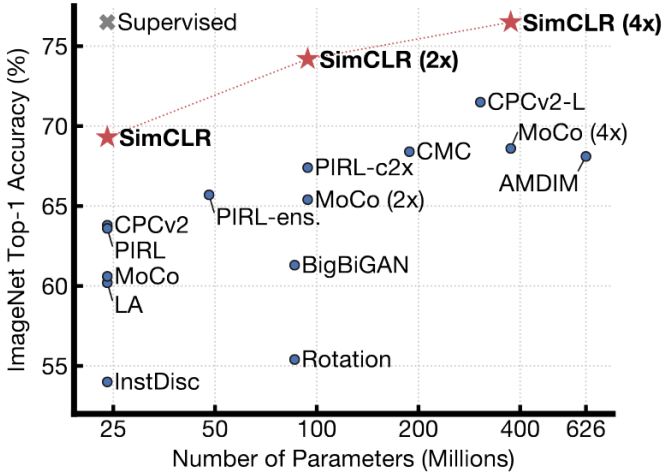
$$\arg\min_f \mathcal{L}(x, y) = \begin{cases} ||f(\mathcal{D} \circ x) - \sum_j f(\mathcal{T}_j \circ x)||_2^2 \\ \max(0, \epsilon - ||f(\mathcal{D} \circ y) - \sum_j f(\mathcal{T}_j \circ x)||_2^2 \end{cases} \quad (7)$$

- **Problems:**
  - Why just using the down-sampling operator $\mathcal{D}$ and the tiling operator $\mathcal{T}_j$?
  - Why using a single "negative" sample y ?
- Take-home message: information preserving *transformations*, *instance discrimination* (without class/prototype), *contrastive loss* are three important ingredients

Summary

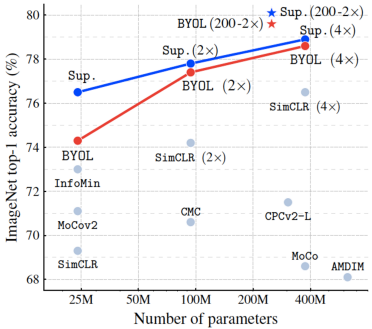
Contrastive Learning

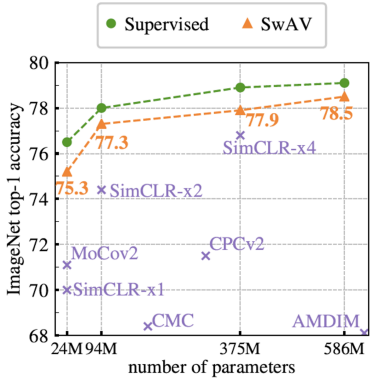
- Contrastive learning methods outperform the other pretext tasks[69]

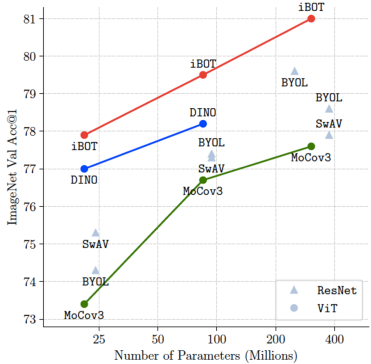


[69] T. Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations". In: *ICML*. 2020.

- And recently there has been a plethora of works about it that is closing the performance gap with supervised pretraining[70,71,72]







[70] J.-B. Grill et al. "Bootstrap your own latent: A new approach to self-supervised Learning". In: *NeurIPS*. 2020.
[71] M. Caron et al. "Unsupervised Learning of Visual Features by Contrasting Cluster Assignments". In: *NeurIPS*. 2020.
[72] J. Zhou et al. "Image BERT Pre-training with Online Tokenizer". In: *ICLR*. 2022.

Contrastive Learning - A bit of history

- **Goal**: given a set of images $x_k \in \mathcal{X}$, learn a mapping function $f_\theta : \mathcal{X} \to \mathcal{F}$ such that:

$$\text{if } x_a \text{ and } x_b \text{ are semantically similar } \rightarrow f(x_a) \approx f(x_b)$$
$$\text{if } x_a \text{ and } x_b \text{ are semantically different } \rightarrow f(x_a) \neq f(x_b)$$

- These conditions can be reformulated from a mathematical point using either a **geometric approach**, based on a distance $d(f(x_a), f(x_b))$, or an **information theoretic approach**, based on a statistical dependence measure, such as Mutual Information $I(f(x_a), f(x_b))$.

$$\text{if } x_a \text{ and } x_b \text{ are semantically similar } \rightarrow$$
$$\arg\min_f d(f(x_a), f(x_b)) \qquad \arg\max_f I(f(x_a), f(x_b))$$

$$\text{if } x_a \text{ and } x_b \text{ are semantically different } \rightarrow$$
$$\arg\max_f d(f(x_a), f(x_b)) \qquad \arg\min_f I(f(x_a), f(x_b))$$

Contrastive Learning - A bit of history

Geometric approach (Y. LeCun)

- Pairwise loss[a]
- Triplet loss[b]
- Tuplet loss[c,d,e]

[a] S. Chopra et al. "Learning a Similarity Metric Discriminatively, with Application to Face Verification". In: *CVPR*. 2005.

[b] F. Schroff et al. "FaceNet: A Unified Embedding for Face Recognition and Clustering". In: *CVPR*. 2015.

[c] H. O. Song et al. "Deep Metric Learning via Lifted Structured Feature Embedding". In: *CVPR*. 2016.

[d] K. Sohn. "Improved Deep Metric Learning with Multi-class N-pair Loss Objective". In: *NIPS*. 2016.

[e] B. Yu et al. "Deep Metric Learning With Tuplet Margin Loss". In: *ICCV*. 2019.

Information theory approach (G. Hinton)

- Soft Nearest Neighbor[a,b]
- Contrastive Predictive Coding (CPC)[c]
- Non-Parametric Instance Discrimination[d]
- Deep InfoMax (DIM)[e]

[a] R. Salakhutdinov et al. "Learning a Nonlinear Embedding by Preserving Class ...". In: *AISTATS*. 2007.

[b] N. Frosst et al. "Analyzing and Improving Representations with the Soft Nearest Neighbor". In: *ICML*. 2019.

[c] A. v. d. Oord et al. *Representation Learning with Contrastive Predictive Coding*. 2018.

[d] Z. Wu et al. "Unsupervised Feature Learning via Non-parametric Instance Discrimination". In: *CVPR*. 2018.

[e] R. D. Hjelm et al. "Learning deep representations by mutual information estimation ...". In: *ICLR*. 2019.

Contrastive Learning - A bit of history

Geometric approach (Y. LeCun)[a]

- Need to define positive (x, x^+) (semantically similar) and negative pairs (x, x^-) (semantically different)
- Need to define similarity measure (or distance) that is maximized (or minimized)
- No constraints/hypotheses about negative samples

[a] S. Chopra et al. "Learning a Similarity Metric Discriminatively, with Application to Face Verification". In: *CVPR*. 2005.

Information theory approach (G. Hinton)[a]

- Need to define **pdf** of positive $(x, x^+) \sim p(x, x^+)$ and negative pairs $(x, x^-) \sim p(x)p(x^-)$ where $x^- \perp\!\!\!\perp x, x^+$
- **Maximize Mutual Information** (I) between positive pairs, given **independent** negative pairs: $I(x; x^+) = I(x; x^+, x^-) = \mathbb{E}_{x^- \sim p(x^-)} I(x; x^+)$[b]
- Need to define an **estimator** of I

[a] S. Becker et al. "Self-organizing neural network that discovers surfaces in random ...". In: *Nature* (1992).

[b] B. Poole et al. "On Variational Bounds of Mutual Information". In: *ICML*. 2019.

Contrastive Learning - A bit of history

- The Information theoretic approach is mathematically sounded and well grounded on the role of Mutual Information (I) estimation in representation learning.

- But ... Large I is **not necessarily predictive** of downstream performance. Good results may depend on architecture choices and inductive biases rather than an accurate I estimation[73]

- Furthermore, a geometric approach:
  - is easy to understand and explain
  - can easily formalize abstract ideas for defining new losses or regularization terms (e.g., data biases)
  - No need of implausible hypothesis (e.g., negative samples independence).

[73] M. Tschannen et al. "On Mutual Information Maximization for Representation Learning". In: *ICLR*. 2020.

Summary

Contrastive Learning - Geometric approach

- Let $x \in \mathcal{X}$ be a sample (*anchor*)
- Let x_i^+ be a similar (positive) sample
- Let x_j^- be a different (negative) sample
- Let P be the number of positive samples
- Let N be the number of negative samples
- Let $f : \mathcal{X} \to \mathbb{S}^{d-1}$ be the mapping
- Let $\mathcal{F} = \mathbb{S}^{d-1}$, a ($d$-1)-sphere

Figure: From Schroff *et al.*[a]

[a] F. Schroff et al. "FaceNet: A Unified Embedding for Face Recognition and Clustering". In: *CVPR*. 2015.

Contrastive Learning - Geometric approach

- Let $x \in \mathcal{X}$ be a sample (*anchor*)
- Let x_i^+ be a similar (positive) sample
- Let x_j^- be a different (negative) sample
- Let P be the number of positive samples
- Let N be the number of negative samples
- Let $f : \mathcal{X} \to \mathbb{S}^{d-1}$ be the mapping
- Let $\mathcal{F} = \mathbb{S}^{d-1}$, a ($d$-1)-sphere

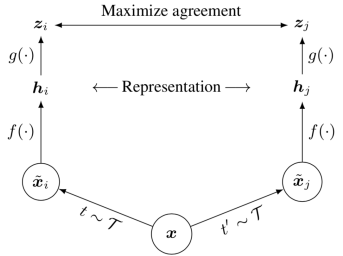
Figure: From Schroff *et al.*[a]

[a] F. Schroff et al. "FaceNet: A Unified Embedding for Face Recognition and Clustering". In: *CVPR*. 2015.

How can we define positive and negative samples ?

Contrastive Learning - Semantic definition

- Most methods use a specific architecture:
  - **Siamese architecture**, initially proposed in 1992/1993 in[74,75], where two networks (the same or related as Teacher/Student) maximize agreement between positive samples and minimize agreement between negative samples
  - Each network is usually divided into two parts: a representation $f()$ and a (non-linear) projector $g()$ network, which improves performance. Why ?



[74] S. Becker et al. "Self-organizing neural network that discovers surfaces in random ...". In: *Nature* (1992).

[75] J. Bromley et al. "Signature Verification using a "Siamese" Time Delay Neural Network". In: *NIPS*. vol. 6. 1993.

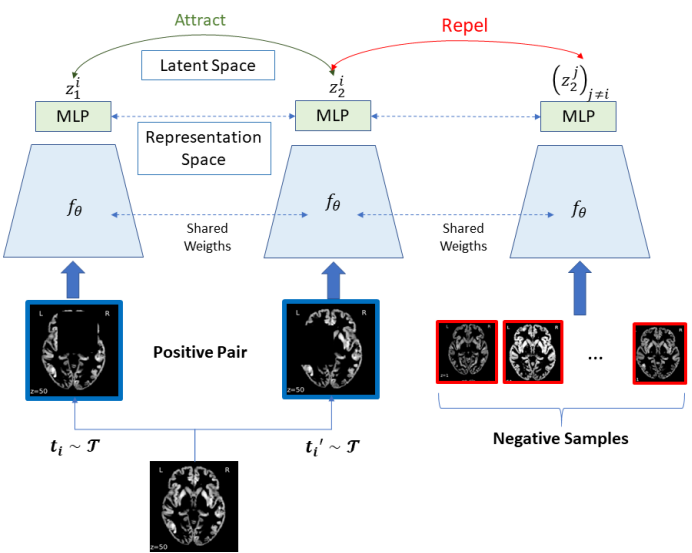
Contrastive Learning - Semantic definition

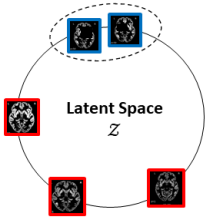
- Positive samples x_i^+ can be defined in different ways:
  - **Unsupervised setting (no label)**: x_i^+ is a transformation of the anchor x[76] or a nearest-neighbor from a support set[77].

[76] T. Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations". In: *ICML*. 2020.

[77] D. Dwibedi et al. "With a Little Help from My Friends: Nearest-Neighbor Contrastive Learning". In: *ICCV*. 2021.

Unsupervised setting





Contrastive Learning - Semantic definition

- Positive samples x_i^+ can be defined in different ways:
  - **Unsupervised setting (no label)**: x_i^+ is a transformation of the anchor x[78] or a nearest-neighbor from a support set[79].
  - **Supervised classification setting (label)**: x_i^+ is a sample belonging to the same class as x.[80]

[78] T. Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations". In: *ICML*. 2020.
[79] D. Dwibedi et al. "With a Little Help from My Friends: Nearest-Neighbor Contrastive Learning". In: *ICCV*. 2021.
[80] P. Khosla et al. "Supervised Contrastive Learning". In: *NeurIPS*. 2020.

Supervised setting

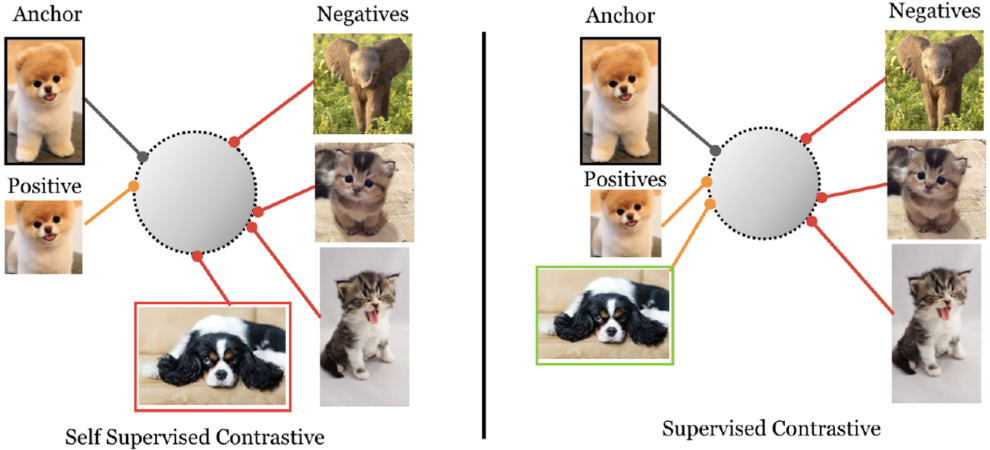


Figure: Image taken from[81]

[81] P. Khosla et al. "Supervised Contrastive Learning". In: *NeurIPS*. 2020.

Contrastive Learning - Semantic definition

- Positive samples x_i^+ can be defined in different ways:
  - **Unsupervised setting (no label)**: x_i^+ is a transformation of the anchor x[82] or a nearest-neighbor from a support set[83].
  - **Supervised classification setting (label)**: x_i^+ is a sample belonging to the same class as x.[84]
  - In **regression**[85] or **weakly-supervised classification**[86]: x_i^+ is a sample with a similar continuous/weak label of x.
- The definition of negative samples x_j^- varies accordingly.

[82] T. Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations". In: *ICML*. 2020.
[83] D. Dwibedi et al. "With a Little Help from My Friends: Nearest-Neighbor Contrastive Learning". In: *ICCV*. 2021.
[84] P. Khosla et al. "Supervised Contrastive Learning". In: *NeurIPS*. 2020.
[85] C. A. Barbano et al. "Contrastive learning for regression in multi-site brain age prediction". In: *IEEE ISBI*. 2023.
[86] B. Dufumier et al. "Contrastive Learning with Continuous Proxy Meta-data for 3D MRI Classification". In: *MICCAI*. 2021.

Contrastive Learning - Semantic definition

- Positive samples x_i^+ can be defined in different ways:
  - **Unsupervised setting (no label)**: x_i^+ is a transformation of the anchor x[82] or a nearest-neighbor from a support set[83].
  - **Supervised classification setting (label)**: x_i^+ is a sample belonging to the same class as x.[84]
  - In **regression**[85] or **weakly-supervised classification**[86]: x_i^+ is a sample with a similar continuous/weak label of x.
- The definition of negative samples x_j^- varies accordingly.

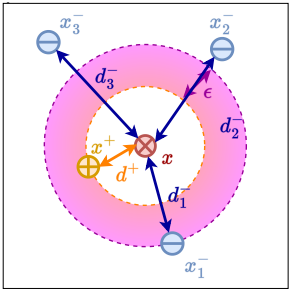
How can we contrast positive and negative samples from a mathematical point of view ?

[82] T. Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations". In: *ICML*. 2020.
[83] D. Dwibedi et al. "With a Little Help from My Friends: Nearest-Neighbor Contrastive Learning". In: *ICCV*. 2021.
[84] P. Khosla et al. "Supervised Contrastive Learning". In: *NeurIPS*. 2020.
[85] C. A. Barbano et al. "Contrastive learning for regression in multi-site brain age prediction". In: *IEEE ISBI*. 2023.
[86] B. Dufumier et al. "Contrastive Learning with Continuous Proxy Meta-data for 3D MRI Classification". In: *MICCAI*. 2021.

Summary

Contrastive Learning - ϵ-margin metric

- We propose to use an ϵ-**margin metric learning** point of view[87]



- If we have a single positive x^+ and several negatives x_j^- (e.g., tuplet loss), we look for f such that:

$$\underbrace{d(f(x), f(x^+))}_{d^+} - \underbrace{d(f(x), f(x_j^-))}_{d_j^-} < -\epsilon \iff \underbrace{s(f(x), f(x_j^-))}_{s_j^-} - \underbrace{s(f(x), f(x^+)}_{s^+} \leq -\epsilon \quad \forall j$$

- where $\epsilon \geq 0$ is a margin between positive and negative, $s(f(a), f(b)) = \langle f(a), f(b) \rangle_2$.

[87] C. A. Barbano et al. "Unbiased Supervised Contrastive Learning". In: *ICLR*. 2023.

Contrastive Learning - ϵ-margin metric

- We propose to use an **ϵ-margin metric learning** point of view[87].
- If we have a single positive x^+ and several negatives x_j^-, we look for f such that:

$$\underbrace{d(f(x), f(x^+))}_{d^+} - \underbrace{d(f(x), f(x_j^-))}_{d_j^-} < -\epsilon \iff \underbrace{s(f(x), f(x_j^-))}_{s_j^-} - \underbrace{s(f(x), f(x^+)}_{s^+} \leq -\epsilon \quad \forall j$$

- where $\epsilon \geq 0$ is a margin between positive and negative, $s(f(a), f(b)) = \langle f(a), f(b) \rangle_2$.
- Two possible ways to transform this Eq. in an optimization problem are:

$$\arg\min_f \max(0, \{s_j^- - s^+ + \epsilon\}_{j=1,..,N}) \qquad \arg\min_f \sum_{j=1}^{N} \max(0, s_j^- - s^+ + \epsilon)$$

- when these losses are $= 0$, the condition is fulfilled. Second is lower-bound of first.

[87] C. A. Barbano et al. "Unbiased Supervised Contrastive Learning". In: *ICLR*. 2023.

Contrastive Learning - ϵ-margin metric

LogSumExp operator *LSE*

The *LogSumExp* operator *LSE* is a smooth approximation of the max function. It is defined as:

$$\max(x_1, x_2, ..., x_N) \leq LSE(x_1, x_2, ..., x_N) = \log(\sum_{i=1}^{N} \exp(x_i))$$

- Using *LSE* with the first problem, we obtain the $\epsilon - InfoNCE$ loss[88]:

$$\arg\min_f \max(0, \{s_j^- - s^+ + \epsilon\}_{j=1,...,N}) \approx \arg\min_f \underbrace{-\log\left(\frac{\exp(s^+)}{\exp(s^+ - \epsilon) + \sum_j \exp(s_j^-)}\right)}_{\epsilon - InfoNCE}$$

[88] C. A. Barbano et al. "Unbiased Supervised Contrastive Learning". In: *ICLR*. 2023.

- When $\epsilon = 0$, we retrieve the InfoNCE[89], whereas when $\epsilon \to \infty$ we obtain the InfoL1O (or Decoupled loss[90])..
- It has been shown[91] that these two losses are lower and upper bound of $I(X^+, X)$:

$$\underbrace{\log \frac{\exp s^+}{\exp s^+ + \sum_j \exp s_j^-}}_{InfoNCE} \leq I(X^+, X) \leq \underbrace{\log \frac{\exp s^+}{\sum_j \exp s_j^-}}_{InfoL1O} \tag{8}$$

- Changing $\epsilon \in [0, \infty)$ can bring to a tighter approximation of $I(X^+, X)$. The exponential function at the denominator $\exp(-\epsilon)$ monotonically decreases as ϵ increases.

[89] A. v. d. Oord et al. *Representation Learning with Contrastive Predictive Coding*. 2018.
[90] C.-H. Yeh et al. "Decoupled Contrastive Learning". In: *ECCV*. 2022.
[91] B. Poole et al. "On Variational Bounds of Mutual Information". In: *ICML*. 2019.

Contrastive Learning - ϵ-margin metric

- The inclusion of multiple positive samples (s_i^+) can lead to different formulations (see[92]). Here, we use the simplest one:

$$s_j^- - s_i^+ \leq -\epsilon \quad \forall i, j$$

$$\sum_i \max(-\epsilon, \{s_j^- - s_i^+\}_{j=1,\dots,N}) \approx \underbrace{-\sum_i \log\left(\frac{\exp(s_i^+)}{\exp(s_i^+ - \epsilon) + \sum_j \exp(s_j^-)}\right)}_{\epsilon-SupInfoNCE} \tag{9}$$

- Another formulation is the SupCon loss[93], which has been presented as the "most straightforward way to generalize" the InfoNCE loss with multiple positive. However...

[92] C. A. Barbano et al. "Unbiased Supervised Contrastive Learning". In: *ICLR*. 2023.

[93] P. Khosla et al. "Supervised Contrastive Learning". In: *NeurIPS*. 2020.

Contrastive Learning - ϵ-margin metric

- ... it actually contains a **non-contrastive constraint**[94] on the positive samples: $s_t^+ - s_i^+ \leq 0 \quad \forall i, t$.

$$s_j^- - s_i^+ \leq -\epsilon \quad \forall i, j \quad \text{and} \quad s_t^+ - s_i^+ \leq 0 \quad \forall i, t \neq i$$

$$\frac{1}{P}\sum_i \max(0, \{s_j^- - s_i^+ + \epsilon\}_j, \{s_t^+ - s_i^+\}_{t \neq i}) \approx \epsilon \underbrace{-\frac{1}{P}\sum_i \log\left(\frac{\exp(s_i^+)}{\sum_t \exp(s_t^+ - \epsilon) + \sum_j \exp(s_j^-)}\right)}_{\epsilon - SupCon}$$

- when $\epsilon = 0$ we retrieve exactly $\mathcal{L}_{out}^{sup}$[95].
- One tries to align all positive samples to a single point in the representation space. Thus losing intra-class variability.

[94] C. A. Barbano et al. "Unbiased Supervised Contrastive Learning". In: *ICLR*. 2023.

[95] P. Khosla et al. "Supervised Contrastive Learning". In: *NeurIPS*. 2020.

Supervised Contrastive Learning - Results

Table: Accuracy on vision datasets. SimCLR and Max-Margin results from[96]. Results denoted with * are (re)implemented with mixed precision due to memory constraints.

Dataset	Network	SimCLR	Max-Margin	SimCLR*	CE*	SupCon*	ϵ-SupInfoNCE*
CIFAR-10	ResNet-50	93.6	92.4	91.74$\pm$0.05	94.73$\pm$0.18	95.64$\pm$0.02	**96.14**$\pm$0.01
CIFAR-100	ResNet-50	70.7	70.5	68.94$\pm$0.12	73.43$\pm$0.08	75.41$\pm$0.19	**76.04**$\pm$0.01
ImageNet-100	ResNet-50	-	-	66.14$\pm$0.08	82.1$\pm$0.59	81.99$\pm$0.08	**83.3**$\pm$0.06

Table: Comparison of ϵ-SupInfoNCE and ϵ-SupCon on ImageNet-100 in terms of top-1 accuracy (%).

Loss	$\epsilon = 0.1$	$\epsilon = 0.25$	$\epsilon = 0.5$
ϵ-SupInfoNCE	**83.25**$\pm$0.39	**83.02**$\pm$0.41	**83.3**$\pm$0.06
ϵ-SupCon	82.83$\pm$0.11	82.54$\pm$0.09	82.77$\pm$0.14

[96] P. Khosla et al. "Supervised Contrastive Learning". In: *NeurIPS*. 2020.

Summary

Unsupervised Learning

- In a supervised o semi-supervised setting, one can use class labels to define positive and negative samples
- In an unsupervised setting, most methods use a Siamese architecture, the (previously presented) InfoNCE loss[97] and the fact that positive samples are defined as transformations of the anchor. Methods mainly differ for implementation choices as:
  - Kind of transformations
  - Negative/Positive selection
  - Prototypes/Dictionaries

[97] A. v. d. Oord et al. *Representation Learning with Contrastive Predictive Coding*. 2018.

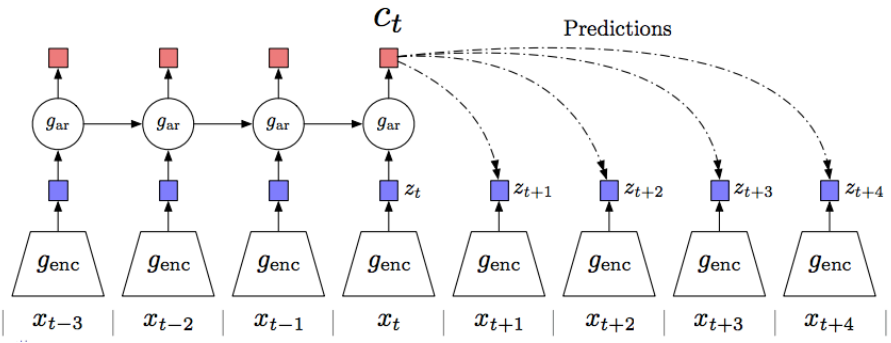
Unsupervised Learning

- In a supervised o semi-supervised setting, one can use class labels to define positive and negative samples
- In an unsupervised setting, most methods use a Siamese architecture, the (previously presented) InfoNCE loss[97] and the fact that positive samples are defined as transformations of the anchor. Methods mainly differ for implementation choices as:
  - Kind of transformations
  - Negative/Positive selection
  - Prototypes/Dictionaries

[97] A. v. d. Oord et al. *Representation Learning with Contrastive Predictive Coding*. 2018.

Kind of transformations

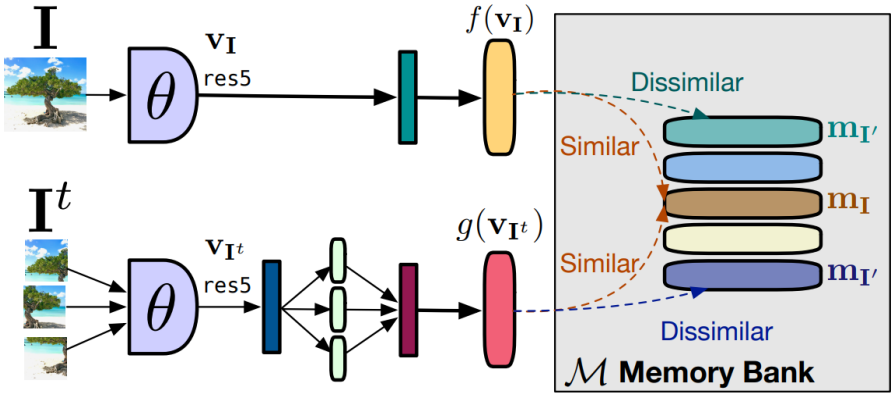
- In Contrastive Predicting Coding (CPCv2)[98] authors combines context prediction and causal modeling by predicting features of patches using only features from patches that lie above. All images are divided into (overlapping) patches in the same way.



[98] O. Henaff et al. "Data-Efficient Image Recognition with Contrastive Predictive Coding". In: *ICML*. 2020.

Kind of transformations

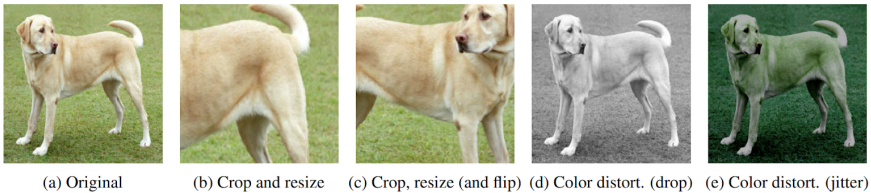
- In PIRL[98] authors use Rotation and Jigsaw as transformations.



[98] I. Misra et al. "Self-Supervised Learning of Pretext-Invariant Representations". In: *CVPR*. 2020.

Kind of transformations

- In SimCLR[98], authors propose a much simpler and more effective (still SOTA) solution using only **random cropping**, **cutout** and **color distortion** (when needed).



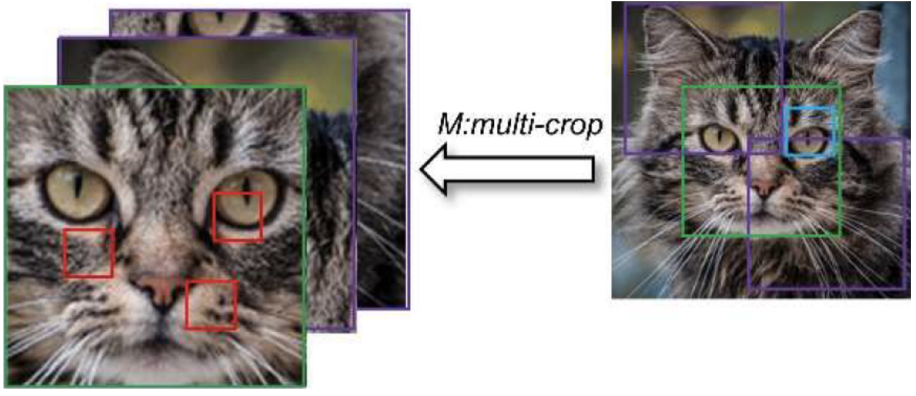
(a) Original (b) Crop and resize (c) Crop, resize (and flip) (d) Color distort. (drop) (e) Color distort. (jitter)

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

[98] T. Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations". In: *ICML*. 2020.

Kind of transformations

- Using multiple cropping (multi-crop[98,99]) with different sizes increases the number of positives without computational over-head



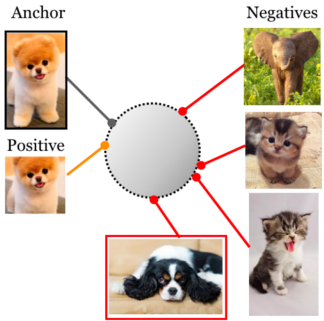
[98] M. Caron et al. "Unsupervised Learning of Visual Features by Contrasting Cluster Assignments". In: *NeurIPS*. 2020.
[99] C. Zhao et al. *Multi-crop Contrastive Learning for Unsupervised Image-to-Image Translation*. 2023.

Unsupervised Learning

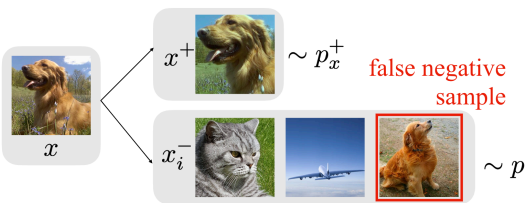
- Kind of transformations
- Negative/Positive selection
- Prototypes/Dictionaries

Negative selection

- Negative selection in an unsupervised setting is very important. Negative samples are usually selected among the other images in the data-set or batch. However, they could actually belong to the same latent class[100,101]



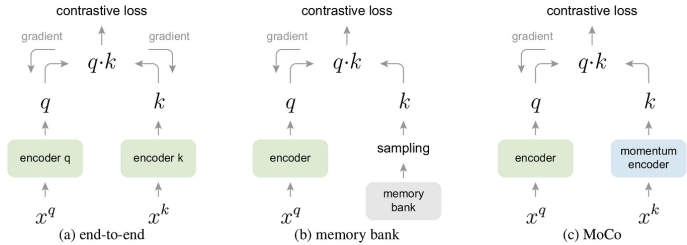
Self Supervised Contrastive



[100] P. Khosla et al. “Supervised Contrastive Learning”. In: *NeurIPS*. 2020.

[101] C.-Y. Chuang et al. “Debiased Contrastive Learning”. In: *NeurIPS*. 2020.

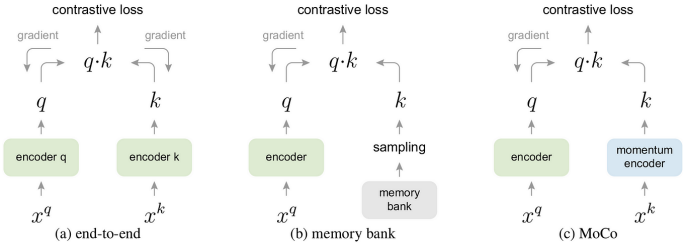
Negative selection



- To limit this issue, since we don't have labels, authors in SimCLR[102] proposed to use very large batch size (and thus very large computational resources) thus increasing the chance of having actual negatives

[102] T. Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations". In: *ICML*. 2020.

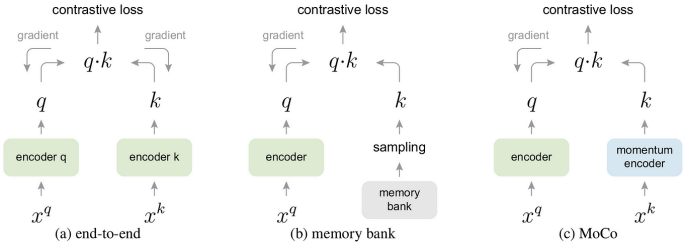
Negative selection



- Large **memory bank**: representations of *all* samples[102] in the data-set or a subset[103] (FIFO queue: old batches replaced by new batches).
- At each iteration, a batch is randomly sampled and updated. The representation of a sample in the bank is updated *only* when it is in the batch, otherwise is kept fixed. Representations updated at different moments are not consistent !

[102] I. Misra et al. "Self-Supervised Learning of Pretext-Invariant Representations". In: *CVPR*. 2020.
[103] K. He et al. "Momentum Contrast for Unsupervised Visual Representation Learning". In: *CVPR*. 2020.

Negative selection



• **Momentum encoder**[102]: updating via back-propagation the representations of the bank is intractable (too large). Use two encoders (siamese), one for the anchor f_q and one for the bank f_k. The parameters of f_q are updated via back-propagating the gradients of the batch and the parameters of f_k using a momentum m. Only parameters and not representations are updated. More memory-efficient and scalable:

$$\theta_k \leftarrow m\theta_k + (1-m)\theta_q \tag{10}$$

[102] K. He et al. "Momentum Contrast for Unsupervised Visual Representation Learning". In: *CVPR*. 2020.

Negative selection

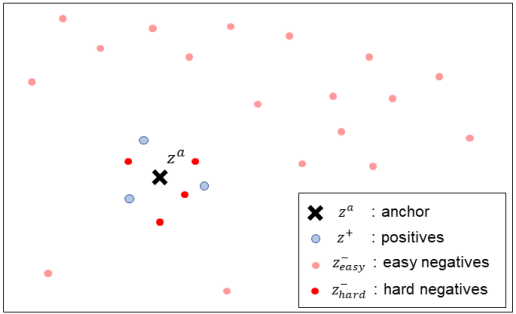
- **Importance sampling techniques** (aka Debiasing), as in[103,104], use an estimator of the True Negatives (TN) based on (user defined) priors about False Negatives (FN) and on several multiple positives (augmentations of the anchor)
- **Decoupled Contrastive Learning (InfoL10 loss)**[105] removes the coupling term (positive term) at the denominator of the InfoNCE loss which hampers performance when:
  1. a positive sample is very close to the anchor
  2. negative samples are far away from the anchor
  3. there is only a small number of negative samples (i.e., a small batch size)

$$\underbrace{\log \frac{\exp s^+}{\exp s^+ + \sum_j \exp s_j^-}}_{\textit{InfoNCE}} \rightarrow \underbrace{\log \frac{\exp s^+}{\sum_j \exp s_j^-}}_{\textit{Decoupled InfoNCE}} \tag{11}$$

[103] C.-Y. Chuang et al. "Debiased Contrastive Learning". In: *NeurIPS*. 2020.
[104] J. Robinson et al. "Contrastive Learning with Hard Negative Samples". In: *ICLR*. 2021.
[105] C.-H. Yeh et al. "Decoupled Contrastive Learning". In: *ECCV*. 2022.

Negative selection

- **Hard negatives** are samples that are mapped nearby the anchor, since they are difficult to distinguish from it, but should be far apart[106]
- In metric learning[107,108], it has been shown that knowing *true* hard negatives can help guide a learning method to correct its mistakes more quickly



[106] T. Jang et al. "Difficulty-Based Sampling for Debiased Contrastive Representation Learning". In: *CVPR*. 2023.
[107] F. Schroff et al. "FaceNet: A Unified Embedding for Face Recognition and Clustering". In: *CVPR*. 2015.
[108] H. O. Song et al. "Deep Metric Learning via Lifted Structured Feature Embedding". In: *CVPR*. 2016.

Negative selection

- Hard negatives can be selected using sampling strategies that are based on the (estimated) similarity at each iteration and that take into account multiple positives[109]
- "Semi-hard negative" can be better since easy negatives (far away) do not help learning and too hard negatives might be outliers/noisy[110]
- The use of a temperature parameter τ in the InfoNCE loss plays a role in controlling the "hardness"[111]. Low τ: focus on the hard negatives. High τ: same importance.

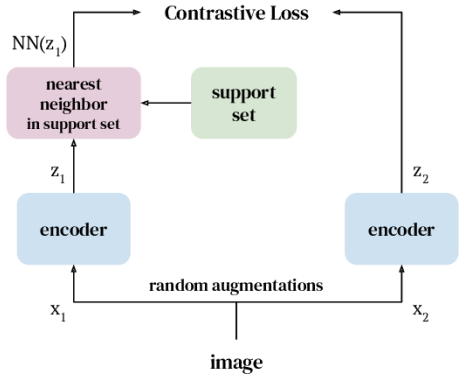
$$\log \frac{\exp\left(s^{+}/\tau\right)}{\exp\left(s^{+}/\tau\right) + \sum_j \exp\left(s_j^{-}/\tau\right)} \tag{12}$$

- Adversarial training can produce more challenging positives and negatives[112]

[109] J. Robinson et al. "Contrastive Learning with Hard Negative Samples". In: *ICLR*. 2021.
[110] C.-Y. Wu et al. "Sampling Matters in Deep Embedding Learning". In: *ICCV*. 2017.
[111] F. Wang et al. "Understanding the Behaviour of Contrastive Loss". In: *CVPR*. 2021.
[112] C.-H. Ho et al. "Contrastive Learning with Adversarial Examples". In: *NeurIPS*. 2020.

Positive selection

- In[113], authors use the nearest-neighbour from a support set (FIFO queue) to sample more positives (and not negatives as before).
- The support set must be big enough, is initialized as a random matrix and is updated using the current batch (as a FIFO queue)



[113] D. Dwibedi et al. "With a Little Help from My Friends: Nearest-Neighbor Contrastive Learning". In: *ICCV*. 2021.

- Building upon the InfoNCE loss, they propose as loss[114]:

$$\mathcal{L}_i^{NNCLR} = -\log \frac{\exp\left(\mathrm{NN}(z_i, Q) \cdot z_i^+ / \tau\right)}{\sum_j \exp\left(\mathrm{NN}(z_i, Q) \cdot z_j / \tau\right)} \tag{13}$$

- where Q is the support set, z_i is the representation of the anchor, z_i^+ is an augmentation of the anchor, z_j is an augmentation of a negative sample and $\mathrm{NN}(z_i, Q)$ is the nearest neighbour of the anchor in the (current) representation space:

$$\mathrm{NN}(z_i, Q) = \underset{q \in Q}{\arg\min} ||z - q||_2 \tag{14}$$

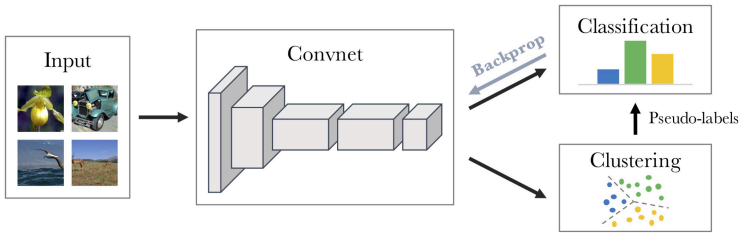
[114] D. Dwibedi et al. "With a Little Help from My Friends: Nearest-Neighbor Contrastive Learning". In: *ICCV*. 2021.

Unsupervised Learning

- Kind of transformations
- Negative/Positive selection
- Prototypes/Dictionaries

Prototypes/Dictionaries

- Combining clustering and self-supervised learning has been lately studied in[115,116]
- They combine two steps:
  1. Given pseudo-labels (cluster assignment Q), minimize cross-entropy (classification)
  2. Cluster data given features of encoder f



[115] M. Caron et al. "Deep Clustering for Unsupervised Learning of Visual Features". In: *ECCV*. 2018.
[116] Y. M. Asano et al. "Self-labelling via simultaneous clustering and representation learning". In: *ICLR*. 2020.

Prototypes/Dictionaries

- We thus want to learn:
  - an encoder f, that map data x to feature vectors
  - a classification head h, which is usually a single linear layer followed by a softmax operator to have class probabilities: $\text{softmax}(h \circ f(x))$
  - a cluster assignment matrix Q
- It can be seen as a EM optimization where we first fix Q and update the parameters of $h \circ f$ and then fix $h \circ f$ and estimate Q
- In[117] authors update Q using the K-means algorithm

[117] M. Caron et al. "Deep Clustering for Unsupervised Learning of Visual Features". In: *ECCV*. 2018.

Prototypes/Dictionaries

- We thus want to learn:
  - an encoder f, that map data x to feature vectors
  - a classification head h, which is usually a single linear layer followed by a softmax operator to have class probabilities: $\text{softmax}(h \circ f(x))$
  - a cluster assignment matrix Q
- It can be seen as a EM optimization where we first fix Q and update the parameters of $h \circ f$ and then fix $h \circ f$ and estimate Q
- In[117] authors update Q using the K-means algorithm $\rightarrow$ Problem: no well-defined objective function and thus no convergence properties
- In[118], authors propose to partition data in equally-sized subsets (optimal transport problem) $\rightarrow$ well defined mathematical framework

[117] M. Caron et al. "Deep Clustering for Unsupervised Learning of Visual Features". In: *ECCV*. 2018.

[118] Y. M. Asano et al. "Self-labelling via simultaneous clustering and representation learning". In: *ICLR*. 2020.

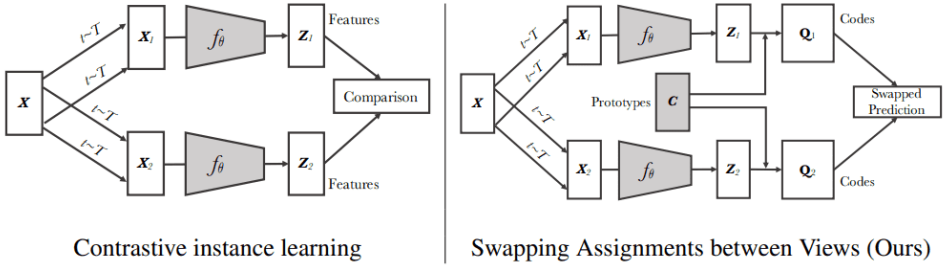
Prototypes/Dictionaries

- Clustering methods have two main problems:
  - They do not scale well with large datasets $\rightarrow$ need a pass over the entire dataset to obtain cluster assignment
  - Number of features of penultimate layer of ConvNet is large $\rightarrow$ need to be PCA-reduced

- To solve this problem one can use prototypes c[119,120]. A code q is obtained for each sample x by mapping its feature $z = f(x)$ to a set of K trainable prototypes $c \rightarrow$ the dimension of q is K, which is much smaller than the one of z

[119] M. Caron et al. "Unsupervised Learning of Visual Features by Contrasting Cluster Assignments". In: *NeurIPS*. 2020.
[120] J. Li et al. "Prototypical Contrastive Learning of Unsupervised Representations". In: *ICLR*. 2021.

Prototypes/Dictionaries

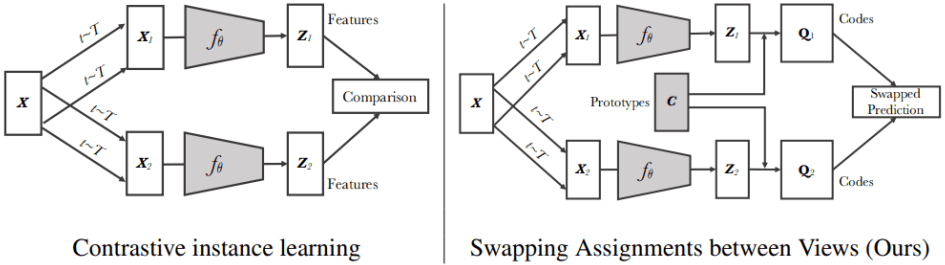


- In SwAV[121], authors propose to compute the codes q_s and q_t of two augmentations of the same image x_{is} and x_{it} and predict each code from the feature z of the other image. Given $z_s = f(x_{is})$ and $z_t = f(x_{it})$, we obtain:

$$\mathcal{L}(z_t, z_s) = \ell(z_t, q_s) + \ell(z_s, q_t) \tag{15}$$

[121] M. Caron et al. "Unsupervised Learning of Visual Features by Contrasting Cluster Assignments". In: *NeurIPS*. 2020

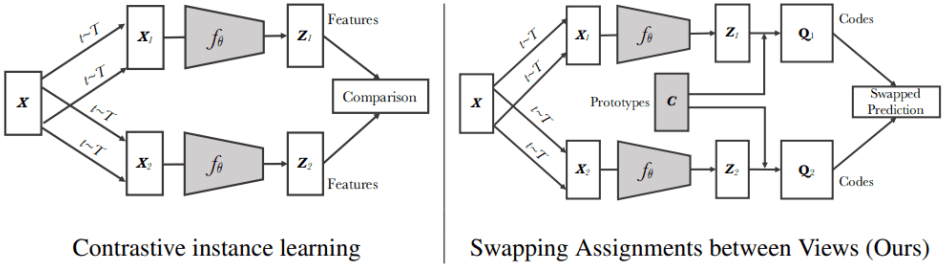
Prototypes/Dictionaries



$$\mathcal{L}(z_t, z_s) = \ell(z_t, q_s) + \ell(z_s, q_t) \tag{16}$$

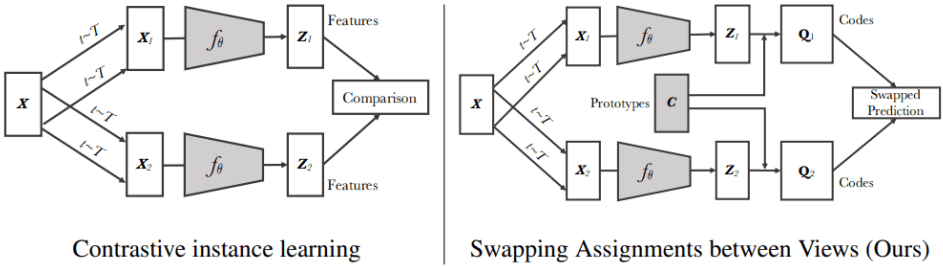
- where $\ell(z, q) = -\sum_k q^k \log(p^k)$ measures the fit between the features z and the code q, namely the cross entropy between the code q^k, corresponding to prototype k, and the probability p^k that the feature z belongs to cluster k.

Prototypes/Dictionaries



- The number of prototypes K is chosen by the user (hyper-parameter) and the soft codes q (no binary assignment) are optimized as to maximize the similarity between the features z and the prototypes c.

Prototypes/Dictionaries



- The number of prototypes K is chosen by the user (hyper-parameter) and the soft codes q (no binary assignment) are optimized as to maximize the similarity between the features z and the prototypes c.

Wait a second... and the negatives !? Swav is actually a non-contrastive method ! We'll talk about that in the next section.

Summary

Contrastive Learning - Weakly supervised

- The previous framework works well when samples are either positive or negative (unsupervised and supervised setting). **But what about continuous/weak labels ?**

- Not possible to determine a hard boundary between positive and negative samples $\rightarrow$ all samples are positive and negative at the same time

Contrastive Learning - Weakly supervised

- The previous framework works well when samples are either positive or negative (unsupervised and supervised setting). **But what about continuous/weak labels ?**
- Not possible to determine a hard boundary between positive and negative samples $\rightarrow$ all samples are positive and negative at the same time
- Let y be the continuous/weak label of the anchor x and y_k of a sample x_k.
- **Simple solution**: threshold d between y and y_k at τ to create positive and negative samples: x_k is x^+ if $d(y, y_k) < \tau \rightarrow$ **Problem**: how to choose τ ?

Contrastive Learning - Weakly supervised

- The previous framework works well when samples are either positive or negative (unsupervised and supervised setting). **But what about continuous/weak labels ?**

- Not possible to determine a hard boundary between positive and negative samples $\rightarrow$ all samples are positive and negative at the same time

- Let y be the continuous/weak label of the anchor x and y_k of a sample x_k.

- **Simple solution**: threshold d between y and y_k at τ to create positive and negative samples: x_k is x^+ is if $d(y, y_k) < \tau \rightarrow$ **Problem**: how to choose τ ?

- **Our solution**: define a degree of "positiveness" between samples using a kernel function $w_k = K_\sigma(y - y_k)$, where $0 \leq w_k \leq 1$.

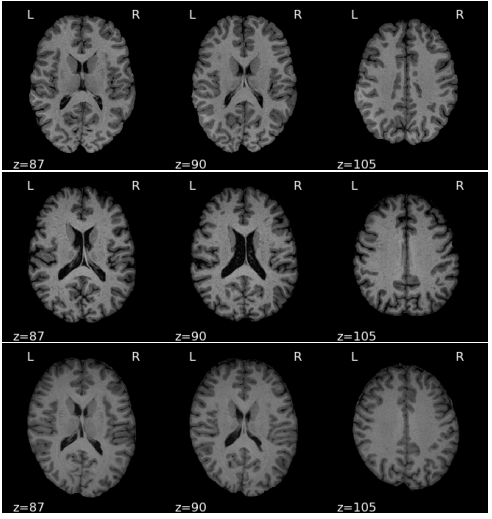
- **New goal**: learn f that maps samples with a high degree of positiveness ($w_k \sim 1$) close in the latent space and samples with a low degree ($w_k \sim 0$) far away from each other.

Question: Which pair of subjects are closer in you opinion (brain MRI, axial plane) ?

Subject A

Subject B

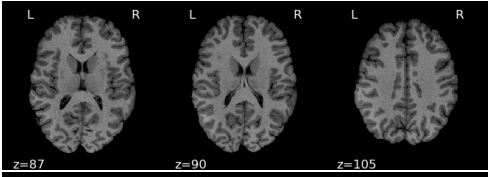
Subject C



Contrastive Learning - Weakly supervised

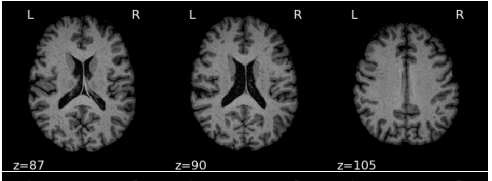
Question: Which pair of subjects are closer in you opinion (brain MRI, axial plane) ?

Subject A



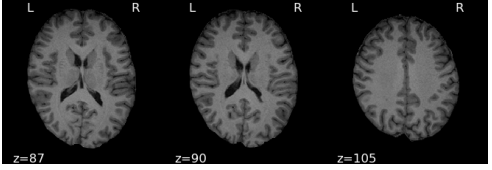
Age=15

Subject B



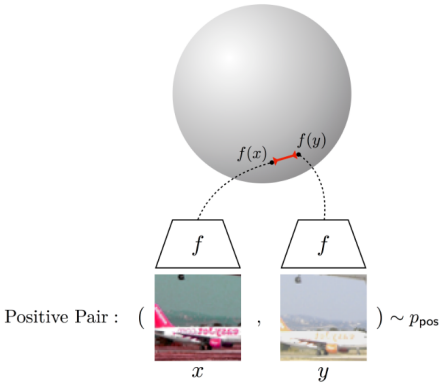
Age=64

Subject C



Age=20

Contrastive Learning - Weakly supervised

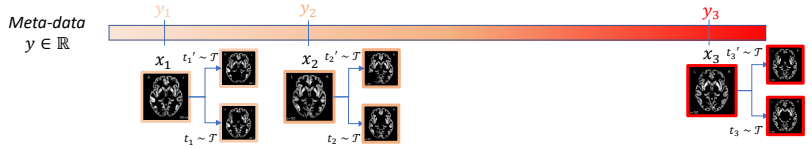


Alignement: $\frac{1}{N}\sum_{i=1}^{N} d_{ii}$

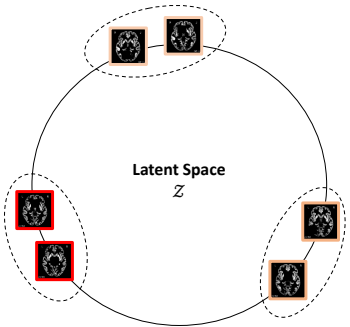
Uniformity: $\log\left(\frac{1}{N^2}\sum_{i,j=1}^{N} e^{-d_{ij}}\right)$[122]

[122] T. Wang et al. "Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere". In: *ICML*. 2020.

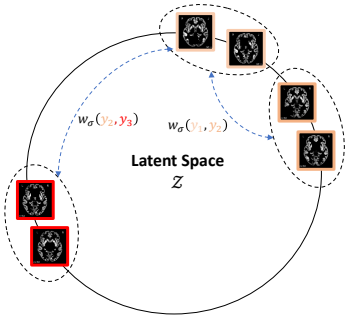
Contrastive Learning - Weakly supervised



SimCLR



y-Aware Contrastive Learning



Contrastive Learning - Weakly supervised

- In[123,124], we propose a new contrastive condition for weakly supervised problems:

$$\frac{w_k}{\sum_j w_j}(s_t - s_k) \leq 0 \quad \forall j, k, t \neq k \in A$$

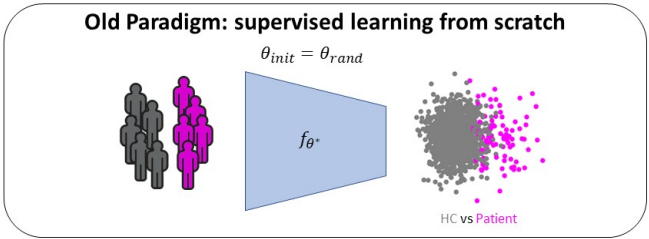
- where A contains the indices of samples $\neq x$ and we consider as positives only the samples with $w_k > 0$, and align them with a strength proportional to w_k.

- As before, we can transform it in an optimization problem obtaining the **y-aware loss**:

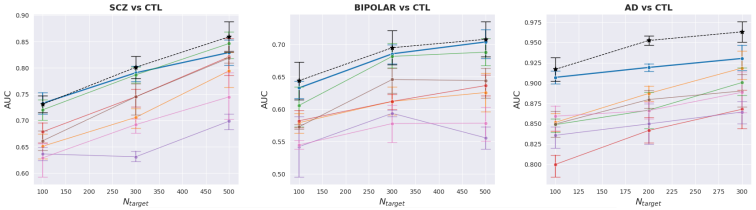
$$\arg\min_f \sum_k \max(0, \frac{w_k}{\sum_t w_t}\{s_t - s_k\}_{\substack{t=1,\dots,N \\ t\neq k}}) \approx \mathcal{L}^{y-aware} = -\sum_k \frac{w_k}{\sum_t w_t} \log\left(\frac{\exp(s_k)}{\sum_{t=1}^{N}\exp(s_t)}\right)$$

[123] B. Dufumier et al. "Contrastive Learning with Continuous Proxy Meta-data for 3D MRI Classification". In: *MICCAI*. 2021.
[124] B. Dufumier et al. "Conditional Alignment and Uniformity for Contrastive Learning...". In: *NeurIPS Workshop*. 2021.

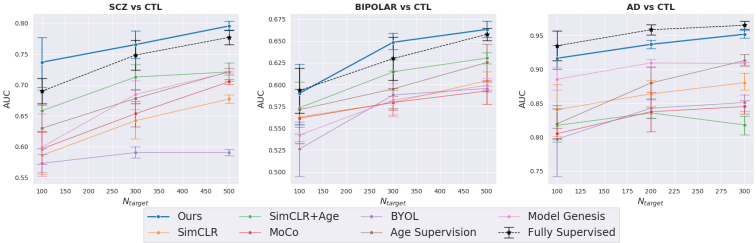
Contrastive Learning - Weakly supervised



Results - Linear evaluation

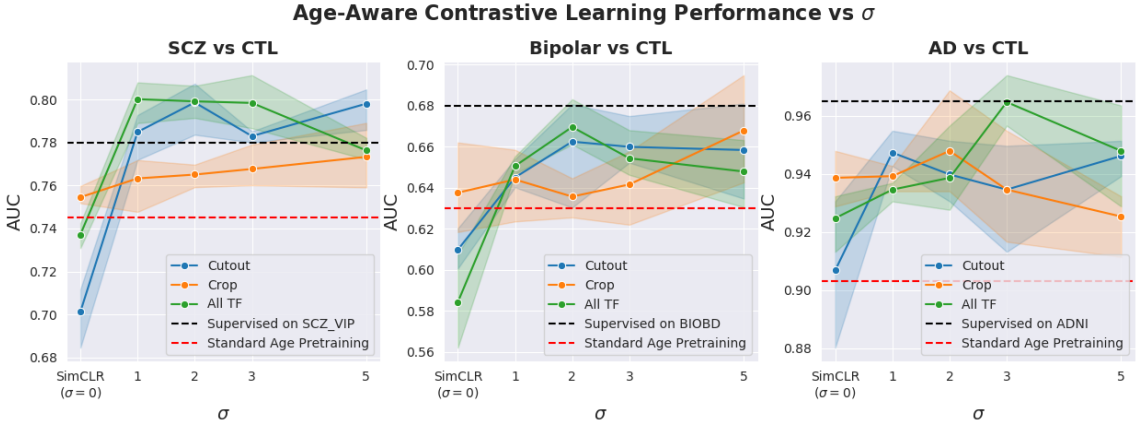


(a) 5-fold CV Stratified on Site.



(b) 5-fold CV Leave-Site-Out

Results - Robustness to σ and transformations



- Linear classification performance remains stable for a range $\sigma \in [1, 5]$
- Adding more transformations improve the representation (in line with SimCLR)
- Cutout remains competitive while being cost-less computationally

Results - Fine-tuning

Task	Test Set		Pre-training Strategies				
			Weakly Self-Supervised	Self-Supervised		Generative	Discriminative
		Baseline	Age-Aware Contrastive[125]	Model Genesis[126]	Contrastive Learning[127]	VAE	Age Sup.
SCZ vs. HC ↑	Internal Test	$85.27_{\pm 1.60}$	**85.17**$_{\pm 0.37}$	$76.31_{\pm 1.77}$	$82.31_{\pm 2.03}$	$82.56_{\pm 0.68}$	$83.05_{\pm 1.36}$
$N_{train} = 933$	External Test	$75.52_{\pm 0.12}$	**77.00**$_{\pm 0.55}$	$67.40_{\pm 1.59}$	$75.48_{\pm 2.54}$	$75.11_{\pm 1.65}$	$74.36_{\pm 2.28}$
BD vs. HC ↑	Internal Test	$76.49_{\pm 2.16}$	**78.81**$_{\pm 2.48}$	$76.25_{\pm 1.48}$	$72.71_{\pm 2.06}$	$71.61_{\pm 0.81}$	$77.21_{\pm 1.00}$
$N_{train} = 832$	External Test	$68.57_{\pm 4.72}$	**77.06**$_{\pm 1.90}$	$65.66_{\pm 0.90}$	$71.23_{\pm 3.05}$	$71.70_{\pm 0.23}$	$73.02_{\pm 2.66}$
ASD vs. HC ↑	Internal Test	$65.74_{\pm 1.47}$	$66.36_{\pm 1.14}$	$63.58_{\pm 4.35}$	$61.92_{\pm 1.67}$	$59.67_{\pm 2.04}$	**67.11**$_{\pm 1.76}$
$N_{train} = 1526$	External Test	$62.93_{\pm 2.40}$	**68.76**$_{\pm 1.70}$	$54.95_{\pm 3.58}$	$61.93_{\pm 1.93}$	$57.45_{\pm 0.81}$	$62.07_{\pm 2.98}$

Table: Fine-tuning results.[128] All pre-trained models use a data-set of **8754 3D MRI of healthy brains**. We reported average AUC(%) for all models and the standard deviation by repeating each experiment three times. Baseline is a DenseNet121 backbone.

[125] B. Dufumier et al. "Contrastive Learning with Continuous Proxy Meta-data for 3D MRI Classification". In: *MICCAI*. 2021.

[126] Z. Zhou et al. "Models Genesis". In: *MedIA* (2021).

[127] T. Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations". In: *ICML*. 2020.

[128] B. Dufumier et al. "Deep Learning Improvement over Standard Machine Learning in Neuroimaging". In: *NeuroImage (under review)* ().

Summary

Contrastive Learning - Regression

- We could use $\mathcal{L}^{y-aware}$ also in regression. But...

$$\mathcal{L}^{y-aware} = -\sum_k \frac{w_k}{\sum_t w_t} \log\left(\frac{\exp(s_k)}{\sum_{t=1}^{N}\exp(s_t)}\right)$$

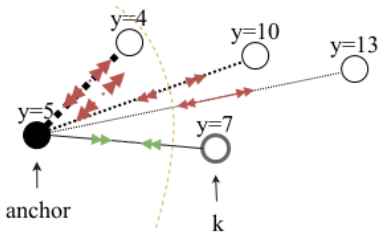
- ... the numerator aligns x_k, and the denominator *focuses more* on the closest samples in the representation space.

Contrastive Learning - Regression

- We could use $\mathcal{L}^{y-aware}$ also in regression. But...

$$\mathcal{L}^{y-aware} = -\sum_k \frac{w_k}{\sum_t w_t} \log\left(\frac{\exp(s_k)}{\sum_{t=1}^{N}\exp(s_t)}\right)$$

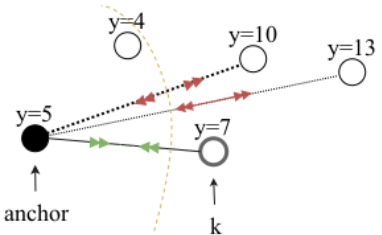
- ... the numerator aligns x_k, and the denominator *focuses more* on the closest samples in the representation space. $\rightarrow$ **Problem !** These samples might have a greater degree of positiveness with the anchor than the considered x_k



- We thus propose two new losses:

$$w_k(s_t - s_k) \leq 0 \quad \text{if } w_t - w_k \leq 0 \quad \forall k, t \neq k \in A(i)$$

$$\mathcal{L}^{thr} = -\sum_k \frac{w_k}{\sum_t \delta_{w_t < w_k} w_t} \log\left(\frac{\exp(s_k)}{\sum_{t \neq k} \delta_{w_t < w_k} \exp(s_t)}\right)$$

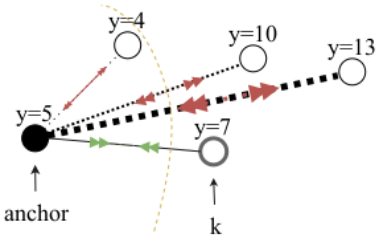


- $\mathcal{L}^{thr}$ repels only the samples that have a y greater than the one of x_k but it still focuses more on the closest samples.

Contrastive Learning - Regression

$$w_k[s_t(1-w_t)-s_k] \leq 0 \quad \forall k, t \neq k \in A(i)$$

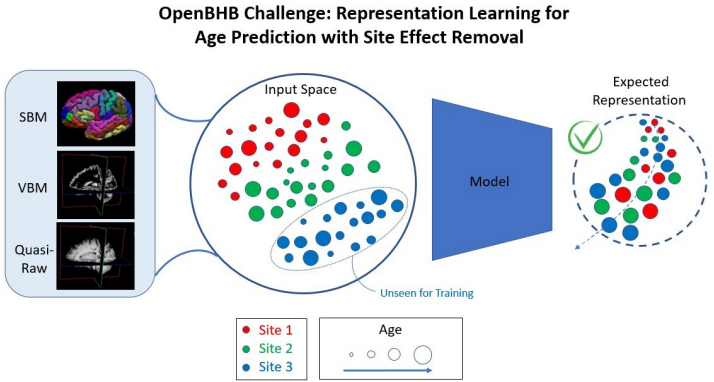
$$\mathcal{L}^{exp} = -\frac{1}{\sum_t w_t} \sum_k w_k \log \frac{\exp(s_k)}{\sum_{t \neq k} \exp(s_t(1-w_t))}$$



- $\mathcal{L}^{exp}$ has a repulsion strength inversely proportional to the similarity between y values, whatever their distance.

- Repulsion strength only depends on the distance in the kernel space. $\rightarrow$ samples close in the kernel space will be close in the representation space.

Results - OpenBHB Challenge



- **OpenBHB Challenge**: age prediction with site-effect removal $\rightarrow$ Brain age $\neq$ chronological age in neurodegenerative disorders !
- N_{train}: 5330 3D brain MRI scans (different subjects) from 71 acquisition sites.
- Two private test data-sets (internal and external)
- To participate `https://ramp.studio/problems/brain_age_with_site_removal`

Results - Regression

Method	Int. MAE ↓	BAcc ↓	Ext. MAE ↓	$\mathcal{L}_c$ ↓
$\mathcal{L}^{y-aware}$	2.66±0.00	6.60±0.17	4.10±0.01	1.82
$\mathcal{L}^{thr}$	2.95±0.01	5.73±0.15	4.10±0.01	1.74
$\mathcal{L}^{exp}$	**2.55**±0.00	**5.1**±0.1	**3.76**±0.01	**1.54**

Table: Comparison of contrastive losses.

Method	Model	Int. MAE ↓	BAcc ↓	Ext. MAE ↓	$\mathcal{L}_c$ ↓
Baseline (ℓ_1)	DenseNet	2.55±0.01	8.0±0.9	7.13±0.05	3.34
	ResNet-18	2.67±0.05	6.7±0.1	4.18±0.01	1.86
	AlexNet	2.72±0.01	8.3±0.2	4.66±0.05	2.21
ComBat	DenseNet	5.92±0.01	2.23±0.06	10.48±0.17	3.38
	ResNet-18	4.15±0.01	**4.5**±0.0	4.76±0.03	1.88
	AlexNet	3.37±0.01	6.8±0.3	5.23±0.12	2.33
$\mathcal{L}^{exp}$	DenseNet	2.85±0.00	5.34±0.06	4.43±0.00	1.84
	ResNet-18	**2.55**±0.00	5.1±0.1	**3.76**±0.01	**1.54**
	AlexNet	2.77±0.01	5.8±0.1	4.01±0.01	1.71

Table: Final scores on the OpenBHB Challenge leaderboard using a 3D ResNet-18. **MAE**: Mean Absolute Error. **BAcc**: Balanced Accuracy for site prediction. **Challenge score**: $\mathcal{L}_c = \text{BAcc}^{0.3} \cdot \text{MAE}_{\text{ext}}$.

The Issue of Biases

- Contrastive learning is more robust than traditional end-to-end approaches, such as cross-entropy, against noise in the data or in the labels[129].

- What about data bias, such as the site-effect ?

Method	Model	Int. MAE ↓	BAcc ↓	Ext. MAE ↓	$\mathcal{L}_c$ ↓
Baseline (ℓ_1)	ResNet-18	2.67±0.05	6.7±0.1	4.18±0.01	1.86
ComBat	ResNet-18	4.15±0.01	4.5±0.0	4.76±0.03	1.88
$\mathcal{L}^{exp}$	ResNet-18	2.55±0.00	5.1±0.1	3.76±0.01	1.54

- $\mathcal{L}^{exp}$ shows a small overfitting on internal sites but also a **low debiasing capability** towards site effect → BAcc should be equal to random chance: $1/n_{\text{sites}} = 1/64 \sim 1.56$

- Need to include debiasing regularization terms, such as FairKL[130]. Please have a look !

[129] F. Graf et al. "Dissecting Supervised Contrastive Learning". In: *ICML*. 2021.

[130] C. A. Barbano et al. "Unbiased Supervised Contrastive Learning". In: *ICLR*. 2023.

Summary

Summary

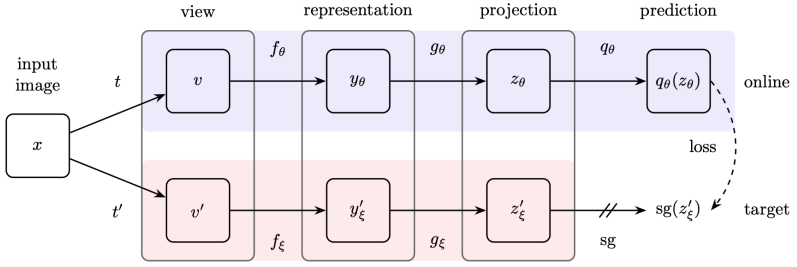
Self-distillation

- Self-distillation methods use a Siamese architecture with *two* different neural networks: online (student) and target (teacher).
- They are non-contrastive methods $\rightarrow \rightarrow$ **only positives, no negatives are used !**
- They avoid collapse using *asymmetric architectures* and *different optimization procedures* (e.g., EMA, stop-gradients, diverse learning rates/weight decay) for the two networks[131,132,133,134]

[131] C. Zhang et al. "How Does SimSiam Avoid Collapse Without Negative Samples?" In: *ICLR*. 2022.
[132] Q. Garrido et al. "On the duality between contrastive and non-contrastive self-supervised". In: *ICLR*. 2023.
[133] Y. Tian et al. "Understanding self-supervised learning dynamics without contrastive pairs". In: *ICML*. 2021.
[134] M. S. Halvagal et al. *Predictor networks and stop-grads provide implicit variance regularization in BYOL/SimSiam*. 2022.

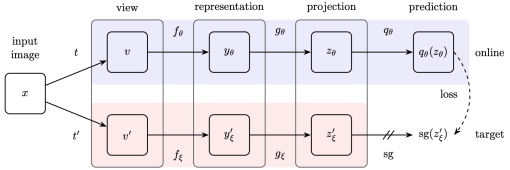
Bootstrap Your Own Latent (BYOL)

- BYOL[135] is the first method that introduced self-distillation to avoid collapse of Siamese architecture (constant output).
- Online is composed of: encoder f_θ, projector g_θ and predictor q_θ.
- Target is composed of: encoder f_ξ and projector g_ξ. The parameters ξ are different from θ and they are updated using an exponential moving average (i.e., momentum encoder): $\xi \leftarrow \tau\xi + (1-\tau)\theta$



[135] J.-B. Grill et al. "Bootstrap your own latent: A new approach to self-supervised Learning". In: *NeurIPS*. 2020.

Bootstrap Your Own Latent (BYOL)

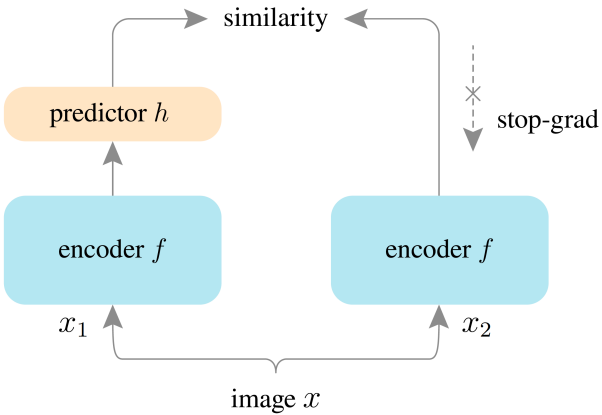


- The idea of BYOL is to maximize agreement between the outputs of the two networks that take as input augmentations of the **same** image
- Agreement is defined as the symmetric mean squared error between normalized outputs (v and v' are fed to both networks):

$$\begin{aligned}\mathcal{L}_{\theta,\xi} &= \frac{1}{2}\left|\left|\frac{q_\theta(z_\theta)}{||q_\theta(z_\theta)||_2} - \frac{z'_\xi}{||z'_\xi||_2}\right|\right|_2^2 + \frac{1}{2}\left|\left|\frac{q_\theta(z'_\theta)}{||q_\theta(z'_\theta)||_2} - \frac{z_\xi}{||z_\xi||_2}\right|\right|_2^2 \\ &\approx -\frac{< q_\theta(z_\theta), z'_\xi >}{||q_\theta(z_\theta)||_2||z'_\xi||_2} - \frac{< q_\theta(z'_\theta), z_\xi >}{||q_\theta(z'_\theta)||_2||z_\xi||_2}\end{aligned} \tag{17}$$

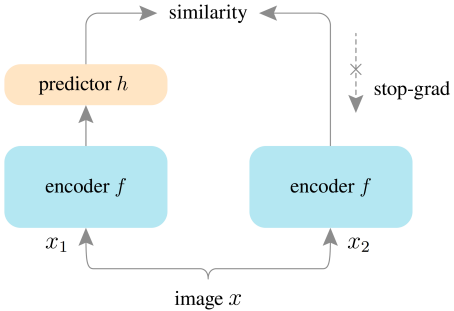
Simple Siamese networks (SimSiam)

- SimSiam[136] removes the EMA (momentum encoder) using simply a stop-grad and... it magically works without collapsing !



[136] X. Chen et al. "Exploring Simple Siamese Representation Learning". In: *CVPR*. 2021.

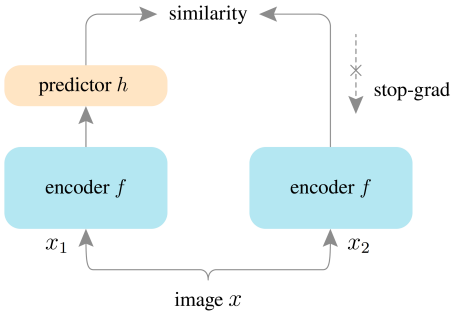
Simple Siamese networks (SimSiam)



- SimSiam uses an even simpler architecture with one *shared* encoder f and a predictor h only for the student network. Two views (i.e., augmentations) of the same image x are then matched minimizing the symmetric loss:

$$\mathcal{L}_{f,h}(x_1, x_2) = -\frac{< h(f(x_1)), f(x_2) >}{||h(f(x_1))||_2 ||f(x_2)||_2} - \frac{< h(f(x_2)), f(x_1) >}{||h(f(x_2))||_2 ||f(x_1)||_2} \tag{18}$$

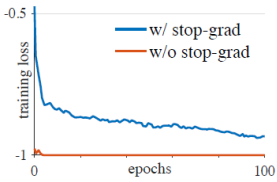
Simple Siamese networks (SimSiam)

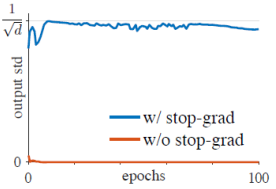


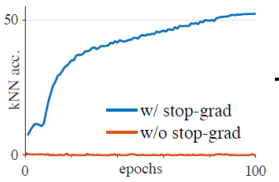
- SimSiam does not collapse thanks to the stop-gradient: $f(x2)$ and $f(x_1)$ are treated as constants in the loss and the encoder f receives no gradients from them

$$\mathcal{L}_{f,h}(x_1, x_2) = -\frac{< h(f(x_1)), f(x_2) >}{||h(f(x_1))||_2 ||f(x_2)||_2} - \frac{< h(f(x_2)), f(x_1) >}{||h(f(x_2))||_2 ||f(x_1)||_2} \tag{19}$$

Simple Siamese networks (SimSiam)







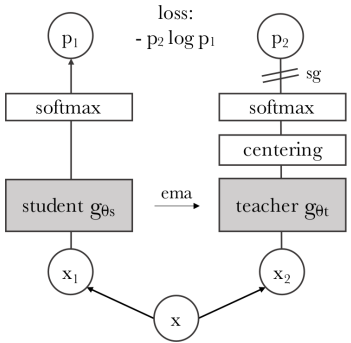
	acc. (%)
w/ stop-grad	67.7±0.1
w/o stop-grad	0.1

	case	proj. MLP's BN hidden	proj. MLP's BN output	pred. MLP's BN hidden	pred. MLP's BN output	acc. (%)
(a)	none	-	-	-	-	34.6
(b)	hidden-only	✓	-	✓	-	67.4
(c)	default	✓	✓	✓	-	68.1
(d)	all	✓	✓	✓	✓	unstable

	pred. MLP h	acc. (%)
baseline	lr with cosine decay	67.7
(a)	no pred. MLP	0.1
(b)	fixed random init.	1.5
(c)	lr not decayed	68.1

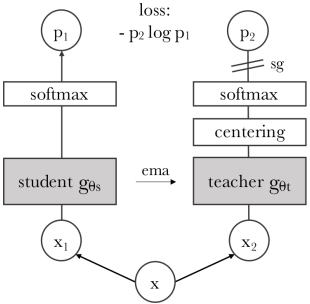
- The stop-gradient and the architecture design (presence of *h*, different learning rates for *f* and *h*, batch normalization, etc) are very important... but, they don't explain **How does SimSiam avoid collapse without negative samples?** → still open question[137]

[137] C. Zhang et al. "How Does SimSiam Avoid Collapse Without Negative Samples?" In: *ICLR*. 2022.



- In[138] authors propose DINO which differs from BYOL and SimSiam: 1) same architecture for teacher and student, 2) knowledge distillation, 3) centering and sharpening the teacher output, 4) use of ViT, 5) multi-crops

[138] M. Caron et al. "Emerging Properties in Self-Supervised Vision Transformers". In: *ICCV*. 2021.



- Teacher and student use the same architecture: $g = f \circ h$ where f is an encoder (ViT or ResNet) and h is a MLP projection head. **No predictor is used for the student!**
- When using ViT no Batch Norms, Group Normalization, Weight decay/standardization are used (thus less hyper-parameters to tune) $\rightarrow$ important for BYOL[139]

[139] P. H. Richemond et al. *BYOL works even without batch statistics*. 2020.

- Generalized knowledge distillation[140] (KD) matches the soft labels of student and teacher instead than hard labels. Soft labels allow to uncover relations between classes that would be difficult to detect with hard labels.
- In DINO the vector of K features z is transformed into soft-labels (each dimension i becomes a probability) using the softmax function:

$$P_s(x)^i = \frac{\exp(g_{\theta_s}(x)^i/\tau_s)}{\sum_{k=1}^{K}\exp(g_{\theta_s}(x)^k/\tau_s)} = \frac{\exp(z_s^i/\tau_s)}{\sum_{k=1}^{K}\exp(z_s^k/\tau_s)}$$
$$P_t(x)^i = \frac{\exp(g_{\theta_t}(x)^i/\tau_t)}{\sum_{k=1}^{K}\exp(g_{\theta_t}(x)^k/\tau_t)} = \frac{\exp(z_t^i/\tau_t)}{\sum_{k=1}^{K}\exp(z_t^k/\tau_t)} \quad (20)$$

- where $z = g(x)^i$ indicates the i-th dimension of z, and τ controls the sharpness of the output distribution (the smaller, the sharper)

[140] D. Lopez-Paz et al. "Unifying distillation and privileged information". In: *ICLR*. 2016.

- Given the parameters of the two networks (θ_t and θ_s), we compute the symmetric loss (as in BYOL and SimSiam):

$$\mathcal{L}_{\theta_s,\theta_t}(x_1, x_2) = \frac{1}{2} H(P_s(x_1), P_t(x_2)) + \frac{1}{2} H(P_s(x_2), P_t(x_1)) \tag{21}$$

- where $H(a, b) = -a \log(b)$ is the cross-entropy.
- As in BYOL, the parameters of the teacher are not updated via back-propagation (i.e., stop-gradient) but using an EMA:

$$\theta_t \leftarrow \lambda \theta_t + (1 - \lambda)\theta_s \tag{22}$$

- Authors use a **multi-crop strategy**: x_1 is a *global* view while x_2 is a *local* view with smaller resolution

- DINO can have two forms of collapse:
  - the model output is uniform along all the dimensions ($z^i = z^j \; \forall \; i, j$)
  - the model output is dominated by one dimension ($z^i >> z^j \; \forall \; j \neq i$)

- DINO can have two forms of collapse:
  - the model output is uniform along all the dimensions ($z^i = z^j \; \forall \, i, j$) $\rightarrow$ sharpening P_t using low τ_t
  - the model output is dominated by one dimension ($z^i >> z^j \; \forall \, j \neq i$) $\rightarrow$ centering P_t

- However, sharpening and centering avoids one collapse but encourage the other, **they need to be used together**

- Centering can be seen as adding a bias term to the teacher: $z_t \leftarrow z_t + c$, and it is updated with an EMA:

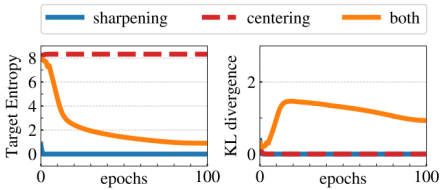
$$c \leftarrow mc + (1 - m)\frac{1}{B}\sum_{i=1}^{B} g_{\theta_t}(x_i) \tag{23}$$

- where $m > 0$ is a rate hyper-parameter and B is the batch size.

	Method	Loss	multi-crop	Center.	BN	Pred.	Top-1
1	DINO	CE	✓	✓			76.1
2	–	MSE	✓	✓			62.4
3	–	CE	✓	✓		✓	75.6
4	–	CE		✓			72.5
5	MoCov2	INCE			✓		71.4
6		INCE	✓		✓		73.4
7	BYOL	MSE			✓	✓	71.4
8	–	MSE			✓		0.1
9	–	MSE		✓			52.6
10	–	MSE	✓		✓	✓	64.8

	Method	Mom.	SK	MC	Loss	Pred.	k-NN	Lin.
1	DINO	✓	✗	✓	CE	✗	72.8	76.1
2		✗	✗	✓	CE	✗	0.1	0.1
3		✓	✓	✓	CE	✗	72.2	76.0
4		✓	✗	✗	CE	✗	67.9	72.5
5		✓	✗	✓	MSE	✗	52.6	62.4
6		✓	✗	✓	CE	✓	71.8	75.6
7	BYOL	✓	✗	✗	MSE	✓	66.6	71.4
8	MoCov2	✓	✗	✗	INCE	✗	62.0	71.6
9	SwAV	✗	✓	✓	CE	✗	64.7	71.8

SK: Sinkhorn-Knopp, MC: Multi-Crop, Pred.: Predictor
CE: Cross-Entropy, MSE: Mean Square Error, INCE: InfoNCE



• **Take home message:** Non-contrastive teacher-student methods do not need negatives and avoid collapse using architectural/optimization solutions. → still not clear what's the best strategy and why it works !

Recap

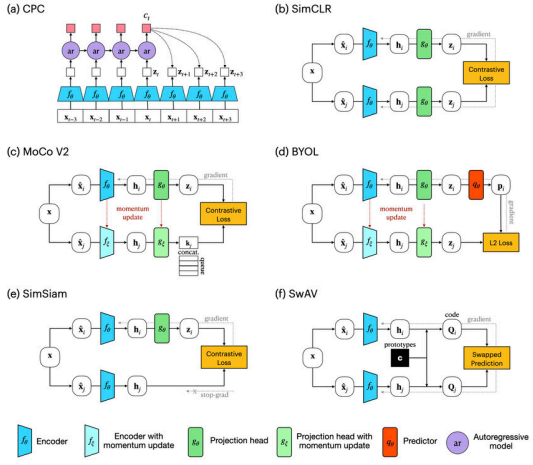


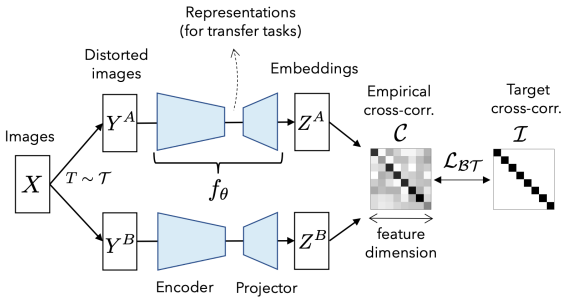
Image from[141]

[141] F. Del Pup et al. *Applications of Self-Supervised Learning to Biomedical Signals: where are we now.* 2023.

Summary

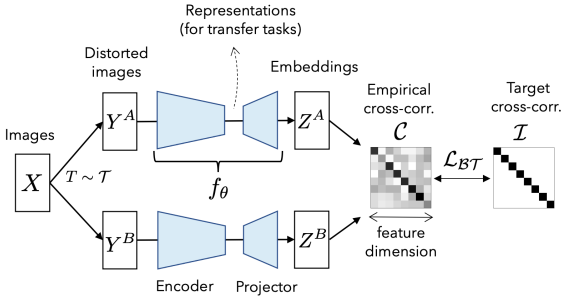
Barlow Twins

- Another category of non-contrastive methods maximizes the information content of the embeddings by reducing the redundancy
- Barlow Twins[142] makes the normalized cross-correlation matrix computed from twin embeddings as close to the identity matrix as possible



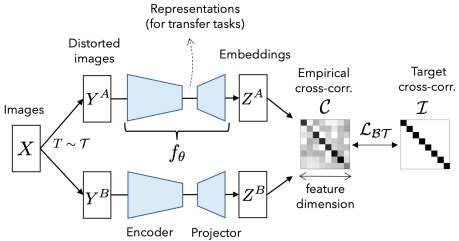
[142] J. Zbontar et al. "Barlow Twins: Self-Supervised Learning via Redundancy Reduction". In: *ICML*. 2021.

Barlow Twins



- Barlow Twins does not use: 1) large batch, 2) asymmetric networks (no predictor), 3) EMA, 4) clustering/prototypes, 5) stop-gradient $\rightarrow$ **Easier to optimize !**
- Two views, y^A and y^B, of the same image x are fed to the same network f producing $z^A = f(y^A)$ and $z^B = f(y^B)$.
- The batch matrix Z with all z (views of multiple images) is then batch-normalized along each dimension i ($\hat{Z}_i = \frac{Z_i - \mu_i}{\sigma_i}$).

Barlow Twins



- We first define the normalized cross-correlation matrix C between the random vectors $\mathbf{z}^A = [\mathbf{z}_1^A, \mathbf{z}_2^A, ..., \mathbf{z}_D^A]^T = [D, 1]$ and $\mathbf{z}^B = [\mathbf{z}_1^B, \mathbf{z}_2^B, ..., \mathbf{z}_D^B]^T = [D, 1]$, where D is the dimension of the embeddings (i.e., number of elements):

$$C = [D, D] \triangleq \mathbb{E}[\mathbf{z}^A (\mathbf{z}^B)^T] = \begin{bmatrix} \mathbb{E}[\mathbf{z}_1^A \mathbf{z}_1^B] & \mathbb{E}[\mathbf{z}_1^A \mathbf{z}_2^B] & ... & \mathbb{E}[\mathbf{z}_1^A \mathbf{z}_D^B] \\ ... & ... & ... & ... \\ \mathbb{E}[\mathbf{z}_D^A \mathbf{z}_1^B] & \mathbb{E}[\mathbf{z}_D^A \mathbf{z}_2^B] & ... & \mathbb{E}[\mathbf{z}_D^A \mathbf{z}_D^B] \end{bmatrix} \tag{24}$$

- We can approximate the $\mathbb{E}$ with an average on the (normalized) batch samples $\hat{Z}^A = [N, D]$ and $\hat{Z}^B = [N, D]$, where N is the batch size, and define $C = ((\hat{Z}^A)^T \hat{Z}^B)/N$.
- Each elements of C has values comprised between -1 (anti-correlation) and 1 (correlation) and is defined:

$$C_{ij} = \frac{\sum_{t=1}^{N} z_{ti}^A z_{tj}^B}{\sqrt{\sum_{t=1}^{N} (z_{ti}^A)^2} \sqrt{\sum_{t=1}^{N} (z_{tj}^B)^2}} \tag{25}$$

- where t is a batch index, i and j refers to the dimension of the embeddings (i.e., $1 \leq i, j \leq D$)

Barlow Twins

- Intuitively, since the two networks A and B take as input two views of the same image, we would like the embeddings z^A and z^B to be **invariant** to the augmentations and thus that homologous elements of z^A and z^B have **similar values** and **same sign**
- At the same time, we would like that our representations z^A and z^B do not contain **redundant information**, which means that different elements of z^A and z^B should have different and unrelated values
- The loss of Barlow Twins is thus:

$$\mathcal{L}_{BT} = \underbrace{\sum_i (1 - C_{ii})^2}_{\text{invariance term}} + \lambda \underbrace{\sum_i \sum_{j \neq i} C_{ij}^2}_{\text{redundancy reduction term}} \tag{26}$$

- where $\lambda > 0$ is a hyper-parameter to tune

Barlow Twins

- Intuitively, since the two networks A and B take as input two views of the same image, we would like the embeddings z^A and z^B to be **invariant** to the augmentations and thus that homologous elements of z^A and z^B have **similar values** and **same sign**
$\rightarrow C_{ii} = 1 \; \forall i$ perfect correlation

- At the same time, we would like that our representations z^A and z^B do not contain **redundant information**, which means that different elements of z^A and z^B should have different and unrelated values
$\rightarrow C_{ij} = 0 \; \forall i, j$ perfect de-correlation

- The loss of Barlow Twins is thus:

$$\mathcal{L}_{BT} = \underbrace{\sum_i (1 - C_{ii})^2}_{\text{invariance term}} + \lambda \underbrace{\sum_i \sum_{j \neq i} C_{ij}^2}_{\text{redundancy reduction term}} \tag{26}$$

- where $\lambda > 0$ is a hyper-parameter to tune

- VICReg[143] is similar to Barlow Twins but instead than using a normalized cross-correlation, they use a loss with three different terms
  - **Mean squared distance** between the embedding vectors z_t^A and z_t^B $\rightarrow$ invariance to the augmentations
  - **Variance regularization term**: the standard deviation (over a batch) of each dimension of the embedding is forced to be above a given threshold. $\rightarrow$ it prevents a collapse (possible in BT) where all dimensions have the same value
  - **Covariance term**: it attracts the off-diagonal coefficients of the covariance matrices of Z^A and Z^B to be close to 0. $\rightarrow$ it reduces redundancy, as in BT, without using cross-correlation $C = ((\hat{Z}^A)^T \hat{Z}^B)/N$ but single covariance matrices $C^A = ((\hat{Z}^A)^T \hat{Z}^A)/N$, $C^B = ((\hat{Z}^B)^T \hat{Z}^B)/N$
- Results between BT and VicReg are very similar...

[143] A. Bardes et al. "VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning". In: *ICLR*. 2022.

Summary

Conclusions

- Different methods of self-supervised learning exist. Recently contrastive and non-contrastive methods have emerged obtaining SOTA results and, in some cases, on-pair or very similar to fully supervised methods
- Architecture (e.g., siamese networks, CNN/ResNet/ViT, BN, temperature τ, etc.) and optimization (e.g., losses, lr, etc.) choices are important. But probably the most important factor in unsupervised SSL is the transformations/augmentations[144] $\rightarrow$ it depends on the downstream task (e.g., classification, recognition, segmentation) and data (e.g., medical, multimodal)
- Recent non-contrastive methods (BYOL, SwAV) do not need negatives (small batch, smaller pre-training datatsets, less memory and less computing power) **BUT**, they can be harder to train (e.g., SimSiam) and their underlying mechanisms are still poorly understood $\rightarrow$ geometric approach as in[145] or[146] ?

[144] I. Bendidi et al. *No Free Lunch in Self Supervised Representation Learning*. 2023.

[145] C. Zhang et al. "How Does SimSiam Avoid Collapse Without Negative Samples?" In: *ICLR*. 2022.

[146] Q. Garrido et al. "On the duality between contrastive and non-contrastive self-supervised". In: *ICLR*. 2023.

Team

Florence Carton

Matthis Maillard

Rebeca Vetil

Emma Sarfati

Camille Ruppli

Giammarco La Barbera

Ali Mammadov

Robin Louiset

Institutions & Partners

UNIVERSITÀ
DI TORINO

References

Asano, Y. M. et al. “Self-labelling via simultaneous clustering and representation learning”. In: *ICLR*. 2020.

Bao, H. et al. “BEiT: BERT Pre-Training of Image Transformers”. In: *ICLR*. 2022.

Barbano, C. A. et al. “Contrastive learning for regression in multi-site brain age prediction”. In: *IEEE ISBI*. 2023.

Barbano, C. A. et al. “Unbiased Supervised Contrastive Learning”. In: *ICLR*. 2023.

Bardes, A. et al. “VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning”. In: *ICLR*. 2022.

Becker, S. et al. “Self-organizing neural network that discovers surfaces in random ...”. In: *Nature* (1992).

Bendidi, I. et al. *No Free Lunch in Self Supervised Representation Learning*. 2023.

Brock, A. et al. “Large Scale GAN Training for High Fidelity Natural Image Synthesis”. In: *ICLR*. 2019.

Bromley, J. et al. “Signature Verification using a ”Siamese” Time Delay Neural Network”. In: *NIPS*. Vol. 6. 1993.

Caron, M. et al. “Deep Clustering for Unsupervised Learning of Visual Features”. In: *ECCV*. 2018.

Caron, M. et al. “Emerging Properties in Self-Supervised Vision Transformers”. In: *ICCV*. 2021.

Caron, M. et al. “Unsupervised Learning of Visual Features by Contrasting Cluster Assignments”. In: *NeurIPS*. 2020.

Chen, M. et al. “Generative Pretraining From Pixels”. In: *ICML*. 2020.

Chen, T. et al. “A Simple Framework for Contrastive Learning of Visual Representations”. In: *ICML*. 2020.

Chen, X. et al. “Exploring Simple Siamese Representation Learning”. In: *CVPR*. 2021.

References

Chopra, S. et al. “Learning a Similarity Metric Discriminatively, with Application to Face Verification”. In: *CVPR*. 2005.

Chuang, C.-Y. et al. “Debiased Contrastive Learning”. In: *NeurIPS*. 2020.

Del Pup, F. et al. *Applications of Self-Supervised Learning to Biomedical Signals: where are we now*. 2023.

Deng, J. et al. “ImageNet: A Large-Scale Hierarchical Image Database”. In: *CVPR*. 2009.

Devlin, J. et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*.

Doersch, C. et al. “Unsupervised Visual Representation Learning by Context Prediction”. In: *ICCV*. 2015.

Donahue, J. et al. “Adversarial Feature Learning”. In: *ICLR*. 2017.

Donahue, J. et al. “Large Scale Adversarial Representation Learning”. In: *NeurIPS*. 2019.

Dosovitskiy, A. et al. “Discriminative Unsupervised Feature Learning with Exemplar CNNs”. In: *IEEE TPAMI* (2016).

Dufumier, B. et al. “Conditional Alignment and Uniformity for Contrastive Learning...”. In: *NeurIPS Workshop*. 2021.

Dufumier, B. et al. “Contrastive Learning with Continuous Proxy Meta-data for 3D MRI Classification”. In: *MICCAI*. 2021.

Dufumier, B. et al. “Deep Learning Improvement over Standard Machine Learning in Neuroimaging”. In: *NeuroImage (under review)* ().

References

Dufumier, B. et al. “OpenBHB: a Large-Scale Multi-Site Brain MRI Data-set for Age Prediction and Debiasing”. In: *NeuroImage* (2022).

Dumoulin, V. et al. “Adversarially Learned Inference”. In: *ICLR*. 2017.

Dwibedi, D. et al. “With a Little Help from My Friends: Nearest-Neighbor Contrastive Learning”. In: *ICCV*. 2021.

Frosst, N. et al. “Analyzing and Improving Representations with the Soft Nearest Neighbor”. In: *ICML*. 2019.

Garrido, Q. et al. “On the duality between contrastive and non-contrastive self-supervised”. In: *ICLR*. 2023.

Gidaris, S. et al. “Unsupervised Representation Learning by Predicting Image Rotations”. In: *ICLR*. 2018.

Goodfellow, I. et al. “Generative Adversarial Nets”. In: *NIPS*. 2014.

Graf, F. et al. “Dissecting Supervised Contrastive Learning”. In: *ICML*. 2021.

Grill, J.-B. et al. “Bootstrap your own latent: A new approach to self-supervised Learning”. In: *NeurIPS*. 2020.

Halvagal, M. S. et al. *Predictor networks and stop-grads provide implicit variance regularization in BYOL/SimSiam*. 2022.

He, K. et al. “Masked Autoencoders Are Scalable Vision Learners”. In: *CVPR*. 2022.

He, K. et al. “Momentum Contrast for Unsupervised Visual Representation Learning”. In: *CVPR*. 2020.

Henaff, O. et al. “Data-Efficient Image Recognition with Contrastive Predictive Coding”. In: *ICML*. 2020.

Hinton, G. E. et al. “Reducing the Dimensionality of Data with Neural Networks”. In: *Science* (2006).

Hjelm, R. D. et al. “Learning deep representations by mutual information estimation ...”. In: *ICLR*. 2019.

References

Ho, C.-H. et al. “Contrastive Learning with Adversarial Examples”. In: *NeurIPS*. 2020.
Ho, J. et al. “Denoising Diffusion Probabilistic Models”. In: *NeurIPS*. 2020.
Jang, T. et al. “Difficulty-Based Sampling for Debiased Contrastive Representation Learning”. In: *CVPR*. 2023.
Khosla, P. et al. “Supervised Contrastive Learning”. In: *NeurIPS*. 2020.
Kingma, D. P. et al. “Auto-Encoding Variational Bayes”. In: *ICLR*. 2014.
Larsson, G. et al. “Colorization as a Proxy Task for Visual Understanding”. In: *CVPR*. 2017.
– .“Learning Representations for Automatic Colorization”. In: *ECCV*. 2016.
Li, J. et al. “Prototypical Contrastive Learning of Unsupervised Representations”. In: *ICLR*. 2021.
Lin, T.-Y. et al. “Microsoft COCO: Common Objects in Context”. In: *ECCV*. 2014.
Littlejohns, T. J. et al. “The UK Biobank imaging enhancement of 100,000 participants:” in: *Nature Communications* (2020).
Lopez-Paz, D. et al. “Unifying distillation and privileged information”. In: *ICLR*. 2016.
Matsoukas, C. et al. “What Makes Transfer Learning Work for Medical Images”. In: *CVPR*. 2022.
Misra, I. et al. “Self-Supervised Learning of Pretext-Invariant Representations”. In: *CVPR*. 2020.
Mustafa, B. et al. *Supervised Transfer Learning at Scale for Medical Imaging*. 2021.
Neyshabur, B. et al. “What is being transferred in transfer learning?” In: *NeurIPS*. 2020.
Noroozi, M. et al. “Representation Learning by Learning to Count”. In: *ICCV*. 2017.

References

Noroozi, M. et al. “Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles”. In: *ECCV*. 2016.

Oord, A. v. d. et al. *Representation Learning with Contrastive Predictive Coding*. 2018.

Pathak, D. et al. “Context Encoders: Feature Learning by Inpainting”. In: *CVPR*. 2016.

Poole, B. et al. “On Variational Bounds of Mutual Information”. In: *ICML*. 2019.

Radford, A. et al. *Language Models are Unsupervised Multitask Learners*.

Radford, A. et al. “Unsupervised Representation Learning with Deep Convolutional GAN”. In: *ICLR*. 2016.

Raghu, M. et al. “Transfusion: Understanding Transfer Learning for Medical Imaging”. In: *NeurIPS*. 2019.

Rezende, D. J. et al. “Variational Inference with Normalizing Flows”. In: *ICML*. 2015.

Richemond, P. H. et al. *BYOL works even without batch statistics*. 2020.

Robinson, J. et al. “Contrastive Learning with Hard Negative Samples”. In: *ICLR*. 2021.

Salakhutdinov, R. et al. “Learning a Nonlinear Embedding by Preserving Class ...”. In: *AISTATS*. 2007.

Schroff, F. et al. “FaceNet: A Unified Embedding for Face Recognition and Clustering”. In: *CVPR*. 2015.

Sohn, K. “Improved Deep Metric Learning with Multi-class N-pair Loss Objective”. In: *NIPS*. 2016.

Song, H. O. et al. “Deep Metric Learning via Lifted Structured Feature Embedding”. In: *CVPR*. 2016.

Thalles Santos, S. *A Short Introduction to Generative Adversarial Networks*. URL: `https://sthalles.github.io`.

Tian, Y. et al. “Understanding self-supervised learning dynamics without contrastive pairs”. In: *ICML*. 2021.

References

Tschannen, M. et al. “On Mutual Information Maximization for Representation Learning”. In: *ICLR*. 2020.
Vincent, P. et al. “Extracting and composing robust features with denoising autoencoders”. In: *ICML*. 2008.
Vincent, P. et al. “Stacked Denoising Autoencoders: Learning Useful Representations ...”. In: *JMLR* (2010).
Wang, F. et al. “Understanding the Behaviour of Contrastive Loss”. In: *CVPR*. 2021.
Wang, T. et al. “Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere”. In: *ICML*. 2020.
Wu, C.-Y. et al. “Sampling Matters in Deep Embedding Learning”. In: *ICCV*. 2017.
Wu, Z. et al. “Unsupervised Feature Learning via Non-parametric Instance Discrimination”. In: *CVPR*. 2018.
Xie, Z. et al. “SimMIM: a Simple Framework for Masked Image Modeling”. In: *CVPR*. 2022.
Yeh, C.-H. et al. “Decoupled Contrastive Learning”. In: *ECCV*. 2022.
Yu, B. et al. “Deep Metric Learning With Tuplet Margin Loss”. In: *ICCV*. 2019.
Zbontar, J. et al. “Barlow Twins: Self-Supervised Learning via Redundancy Reduction”. In: *ICML*. 2021.
Zhang, C. et al. “How Does SimSiam Avoid Collapse Without Negative Samples?” In: *ICLR*. 2022.
Zhang, R. et al. “Colorful Image Colorization”. In: *ECCV*. 2016.
– .“Split-Brain Autoencoders: Unsupervised Learning by Cross-Channel Prediction”. In: *CVPR*. 2017.
Zhao, C. et al. *Multi-crop Contrastive Learning for Unsupervised Image-to-Image Translation*. 2023.
Zhou, J. et al. “Image BERT Pre-training with Online Tokenizer”. In: *ICLR*. 2022.

References

Zhou, Z. et al. “Models Genesis”. In: *MedIA* (2021).