OpenAI

Self-Supervised Learning

Self-Prediction and Contrastive Learning

Lilian Weng, Jong Wook Kim

NeurIPS 2021 Tutorial

Outline

- Introduction: motivation, basic concepts, examples.
- Early work: look into connection with old methods.
- Methods
  - Self-prediction
  - Contrastive Learning
- Pretext tasks: a wide range of literature review.
- Techniques: improve training efficiency.
- Future directions

Introduction

What is self-supervised learning and why we need it?

What is Self-Supervised Learning?

Self-Supervised Learning (SSL) is a special type of representation learning that enables learning good data representation from unlabelled dataset.

It is motivated by the idea of *constructing supervised learning tasks out of unsupervised datasets.*

What is Self-Supervised Learning?

Self-Supervised Learning (SSL) is a special type of representation learning that enables learning good data representation from unlabelled dataset.

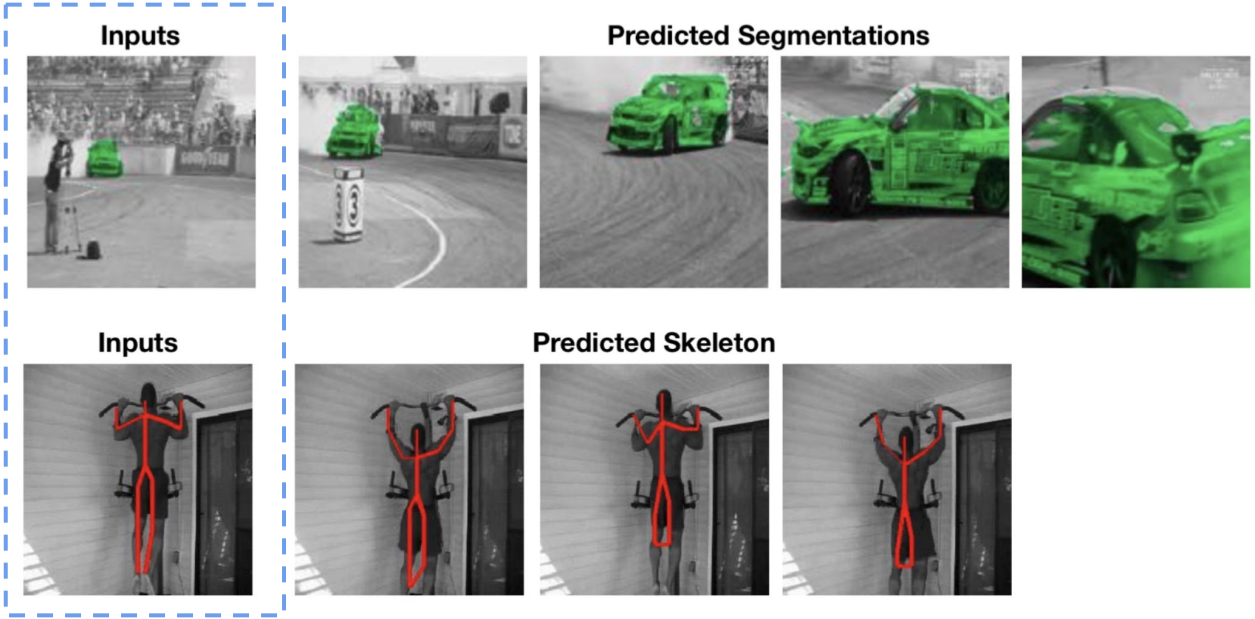
It is motivated by the idea of *constructing supervised learning tasks out of unsupervised datasets.* **Why?**

1. Data labeling is expensive and thus high-quality labeled dataset is limited.
2. Learning good representation makes it easier to transfer useful information to a variety of downstream tasks.
    - e.g. A downstream task has only a few examples.
    - e.g. Zero-shot transfer to new tasks.

Self-supervised learning tasks are also known as ***pretext tasks***.

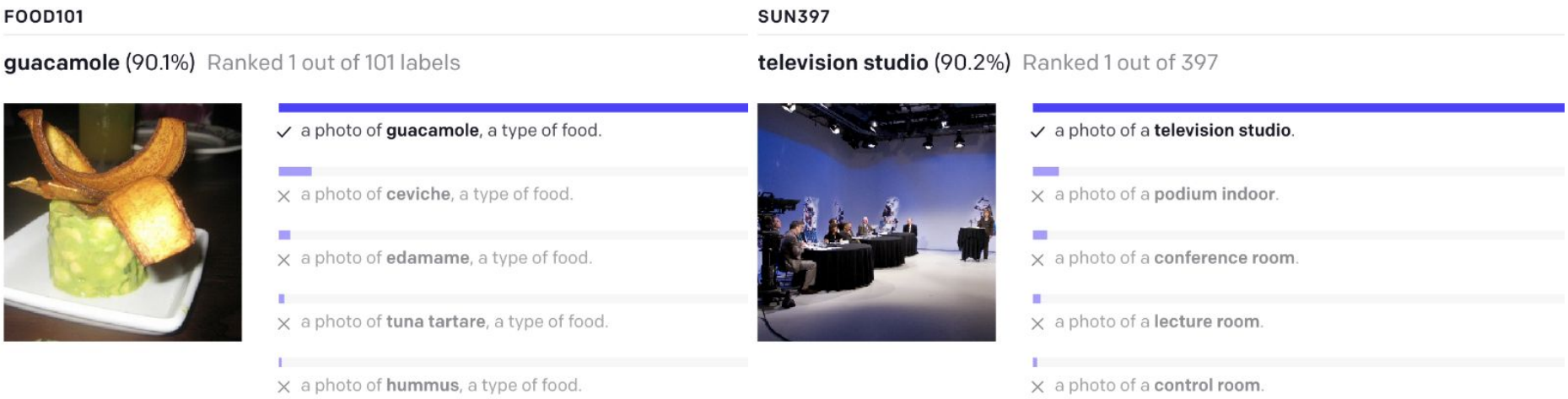
What's Possible with Self-Supervised Learning?

Video colorization (Vondrick et al 2018), as a self-supervised learning method, resulting in a rich representation that can be used for video segmentation and unlabelled visual region tracking, without extra fine-tuning.



What's Possible with Self-Supervised Learning?

Despite of not training on supervised labels, the zero-shot CLIP (Radford et al. 2021) classifier achieve great performance on challenging image-to-text classification tasks.

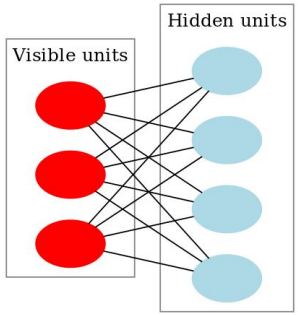


Early Work

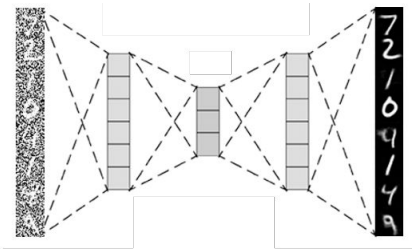
Precursors to recent self-supervised approaches

Early Work: Connecting the Dots

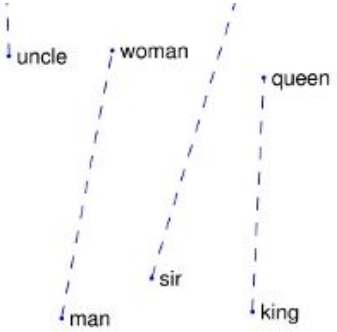
Restricted Boltzmann Machines



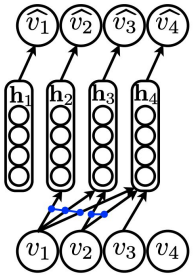
Autoencoders



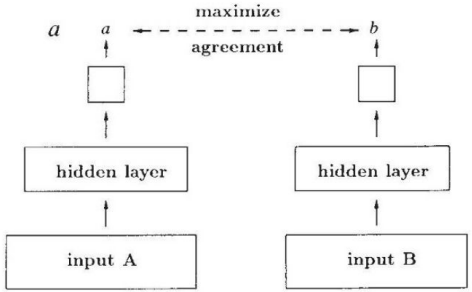
Word2Vec



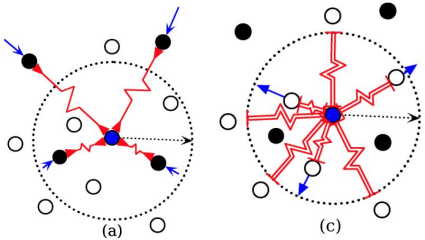
Autoregressive Modeling



Siamese networks

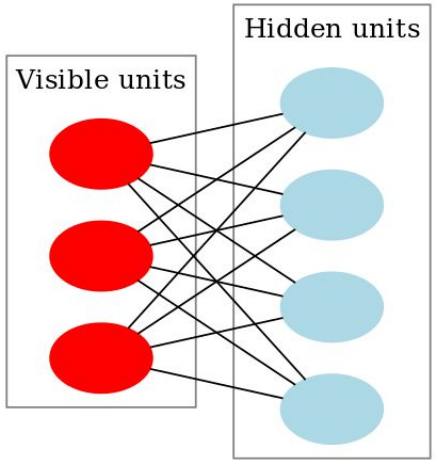


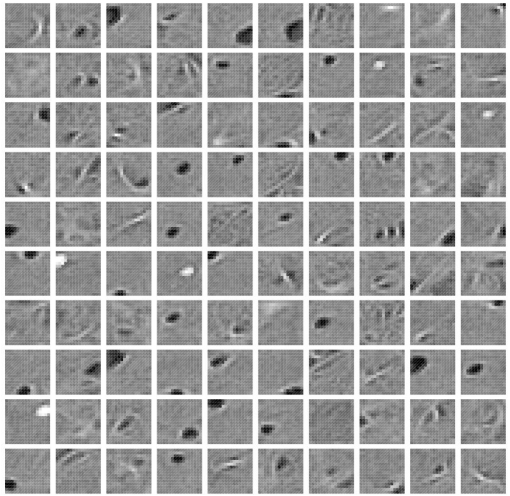
Multiple Instance/Metric Learning



Restricted Boltzmann Machines

- Harmonium (Smolensky 1986)
- Contrastive divergence (Hinton 2000; Hinton 2002)
- Greedy layer-wise pre-training (Hinton et al. 2006; Bengio et al. 2007)

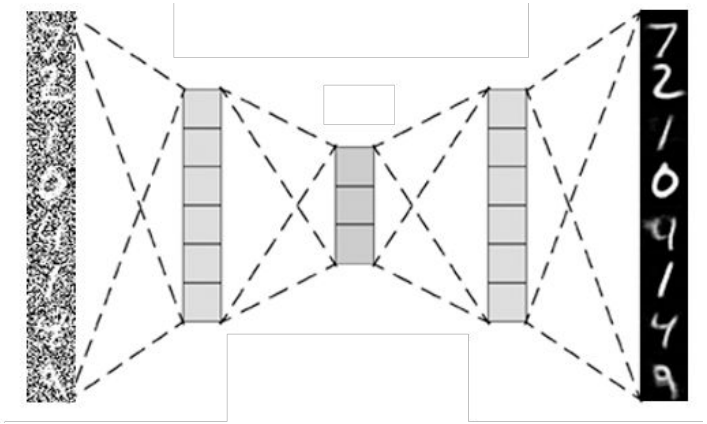


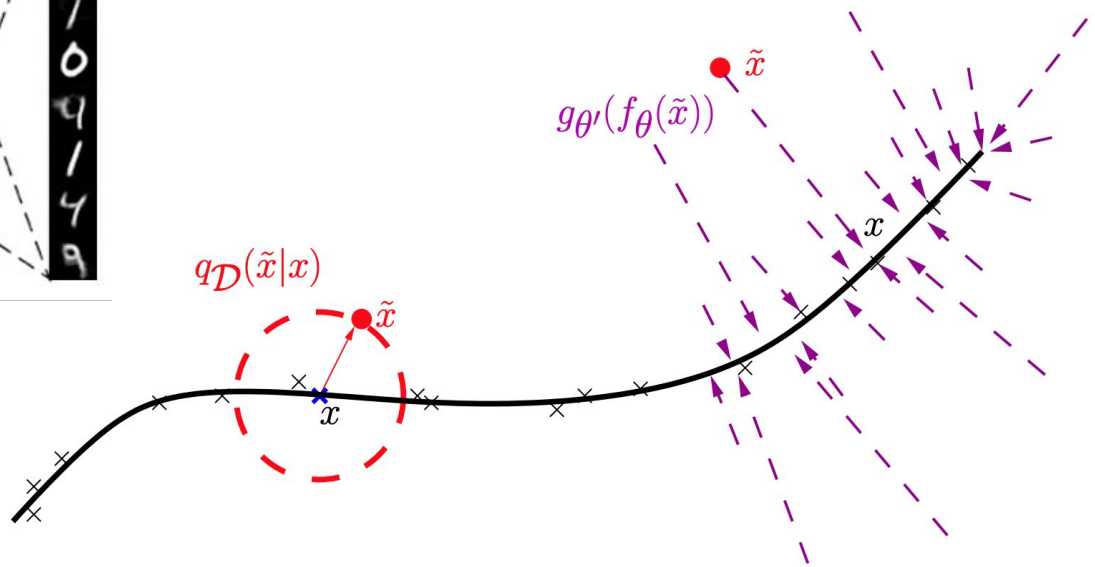


(Hinton 2000)

Autoencoder: Self-Supervised Learning for Vision in Early Days

Denoising Autoencoder (Vincent et al. 2008)





(Vincent et al. 2008)

Word2Vec: Self-Supervised Learning for Language

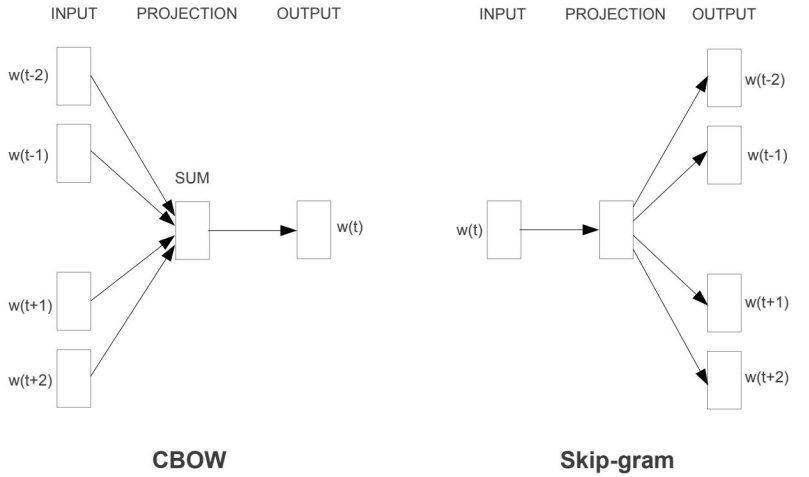
Word embeddings to map words to vectors

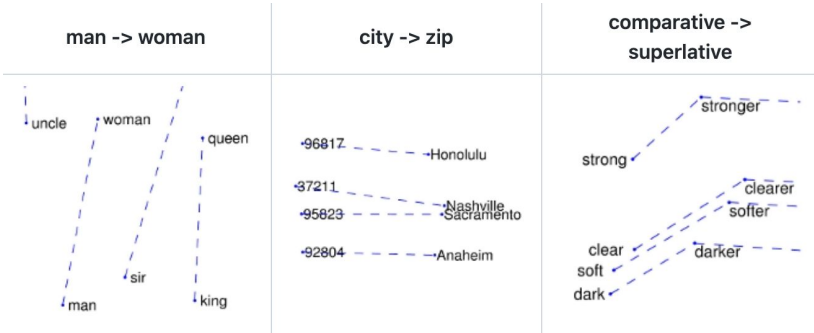
CBOW & Skip-gram (Mikolov et al. 2013)

- Neighboring words → middle word (CBOW)
- Word → neighboring words (skip-gram)

GloVe (Pennington et al. 2014)

- Log-bilinear on word co-occurrences





(Mikolov et al. 2013; Pennington et al. 2014)

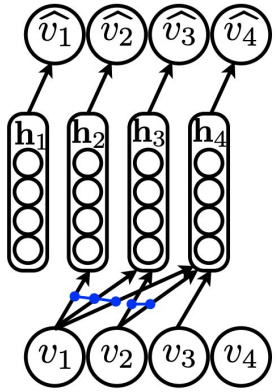
Autoregressive Modeling

$$p(\mathbf{v}) = \prod_{i=1}^{D} p(v_i | \mathbf{v}_{<i})$$

Hidden Markov Models (Baum & Petrie 1966)

Recurrent Neural Networks (Williams, Hinton, & Rumelhart 1986)

Neural Autoregressive Distribution Estimator (Larochelle et al. 2011)

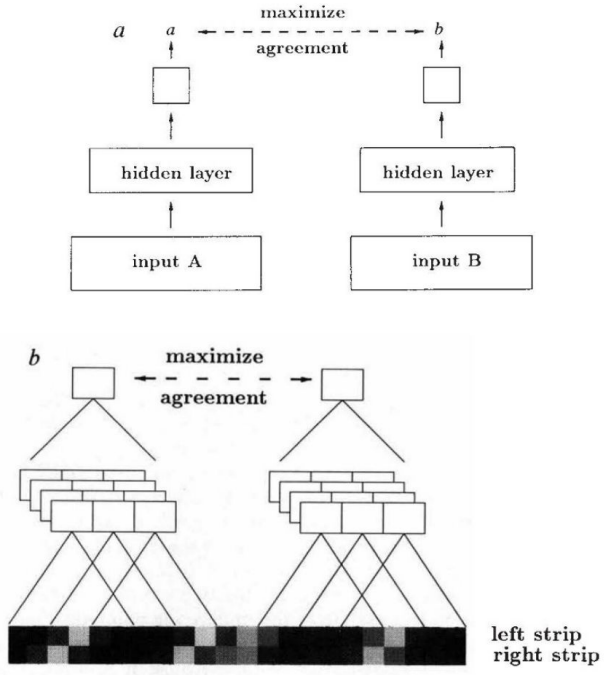


NADE

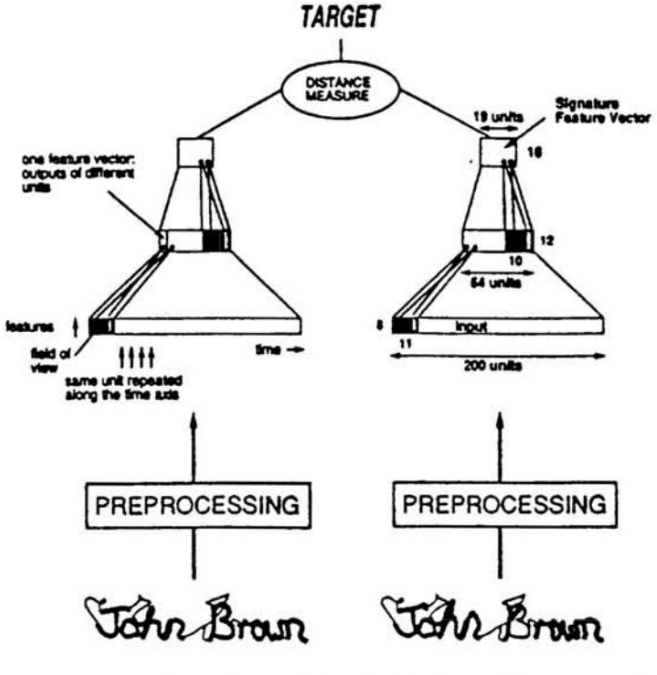
(Larochelle et al. 2011)

Siamese Networks

Self-organizing neural networks
(Becker & Hinton 1992)



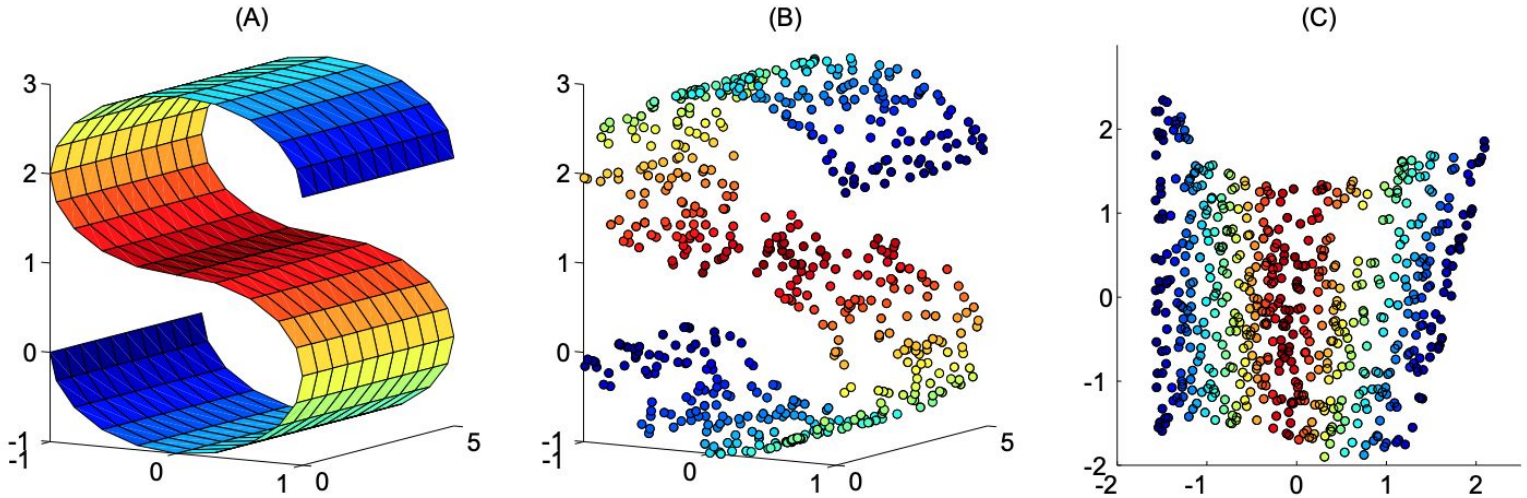
Siamese networks
Bromley et al. (1994)



Multiple Instance Learning & Metric Learning

Multidimensional scaling (MDS; Cox et al. 1994)

Locally linear embedding (LLE; Roweis et al. 2000)



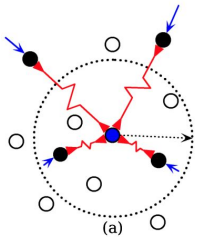
Multiple Instance Learning & Metric Learning

Metric learning (Xing et al. 2002)

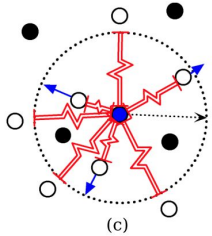
$$d_A(x, y) = ||x - y||_A = \sqrt{(x - y)^T A (x - y)}$$

Contrastive Loss (Chopra & Hadsell et al. 2005)

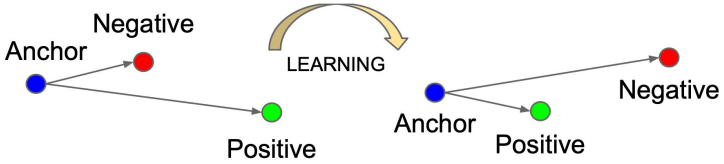
i. If $Y_{ij} = 0$, then update W to decrease $D_W = \|G_W(\vec{X}_i) - G_W(\vec{X}_j)\|_2$



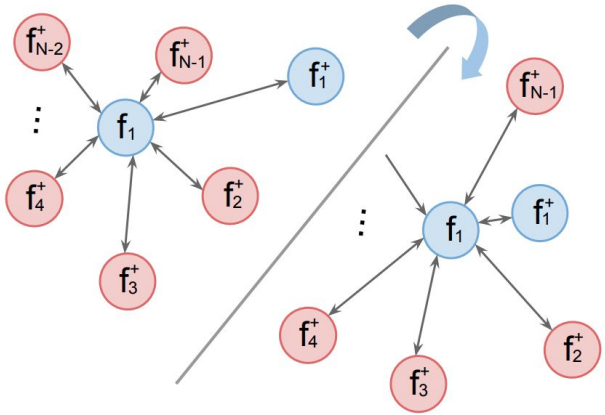
ii. If $Y_{ij} = 1$, then update W to increase $D_W = \|G_W(\vec{X}_i) - G_W(\vec{X}_j)\|_2$



Triplet loss (Schroff et al. 2015)

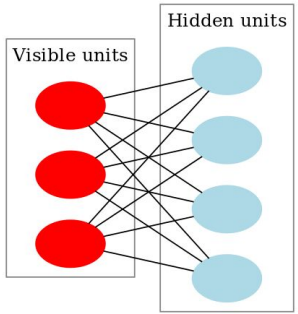


N-pair loss (Sohn 2016)

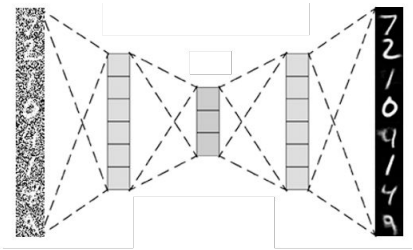


Early Work: Connecting the Dots

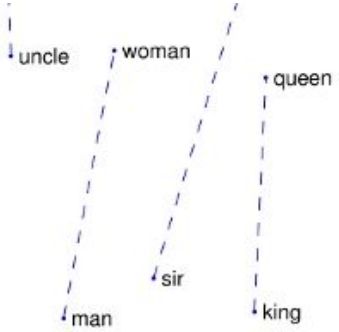
Restricted Boltzmann Machines



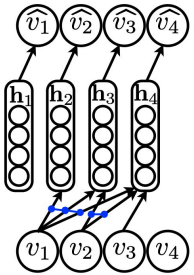
Autoencoders



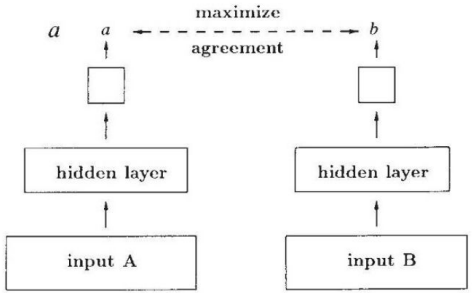
Word2Vec



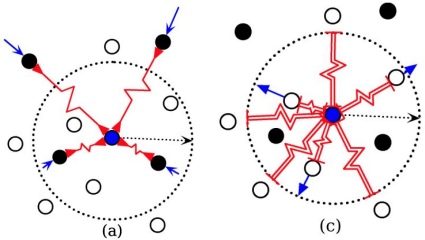
Autoregressive Modeling



Siamese networks



Multiple Instance/Metric Learning



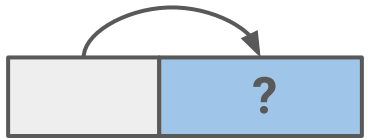
Methods

- Self-prediction
- Contrastive learning

Methods for Framing Self-Supervised Learning Tasks

Self-prediction: Given an individual data sample, the task is to predict one part of the sample given the other part.

The part to be predicted pretends to be missing.

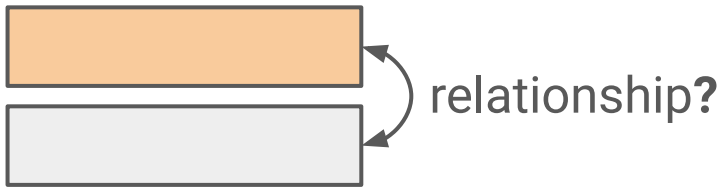


“Intra-sample” prediction

Methods for Framing Self-Supervised Learning Tasks

Contrastive learning: Given multiple data samples, the task is to predict the relationship among them.

The multiple samples can be selected from the dataset based on some known logics (e.g. the order of words / sentences), or fabricated by altering the original version.

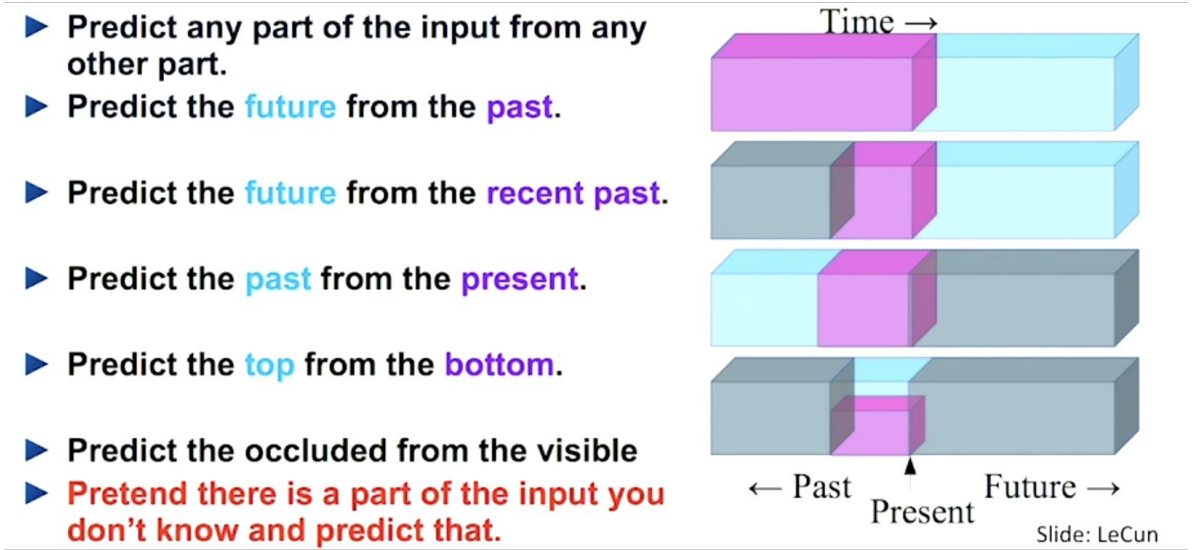


“Inter-sample” prediction

Methods: Self-Prediction

Self-Prediction

Self-prediction construct prediction tasks within every individual data sample: to predict a part of the data from the rest while pretending we don't know that part.



(Famous illustration from Yann LeCun)

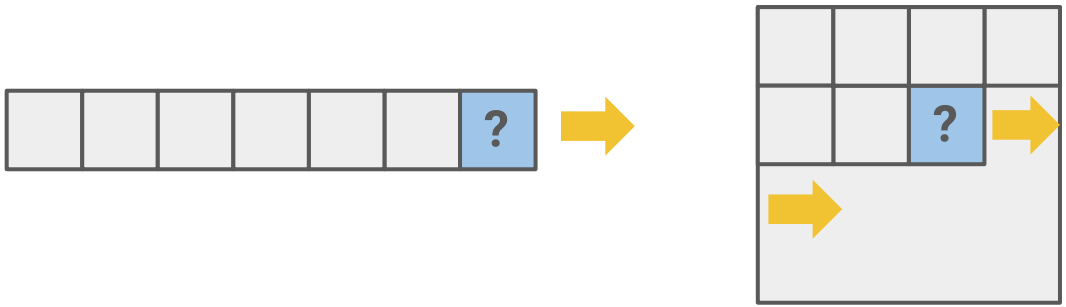
Self-Prediction

Self-prediction construct prediction tasks within every individual data sample: to predict a part of the data from the rest while pretending we don’t know that part.

1. Autoregressive generation
2. Masked generation
3. Innate relationship prediction
4. Hybrid self-prediction

Self-Prediction: Autoregressive Generation

The autoregressive model predicts future behavior based on past behavior. Any data that comes with an innate sequential order can be modeled with regression.

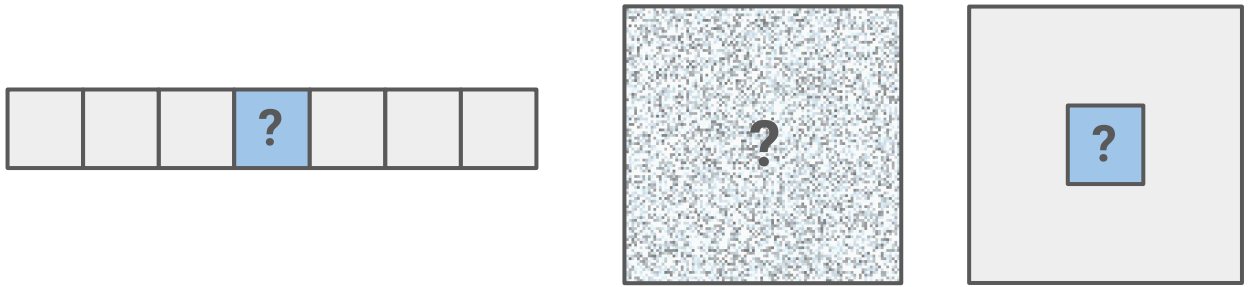


Examples:

- Audio (WaveNet, WaveRNN)
- Autoregressive language modeling (GPT, XLNet)
- Images in raster scan (PixelCNN, PixelRNN, iGPT)

Self-Prediction: Masked Generation

We mask a random portion of information and pretend it is missing, irrespective of the natural sequence. The model learns to predict the missing portion given other unmasked information.

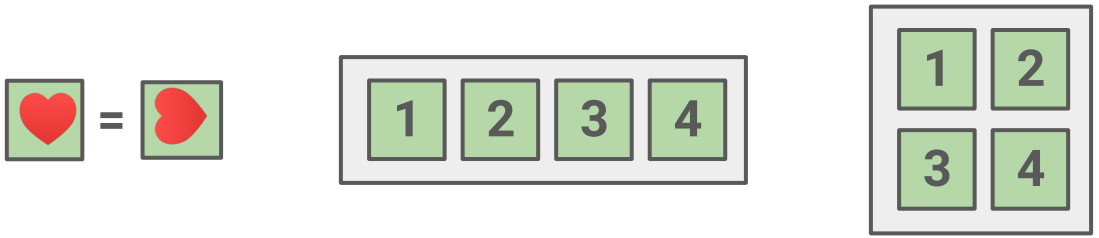


Examples:

- Masked language modeling (BERT)
- Images with masked patch (denoising autoencoder, context autoencoder, colorization)

Self-Prediction: Innate Relationship Prediction

Some transformation (e.g. segmentation, rotation) of one data sample should maintain the original information or follow the desired innate logic.



Examples:

- Order of image patches (e.g., relative position, jigsaw puzzle)
- Image rotation
- Counting features across patches

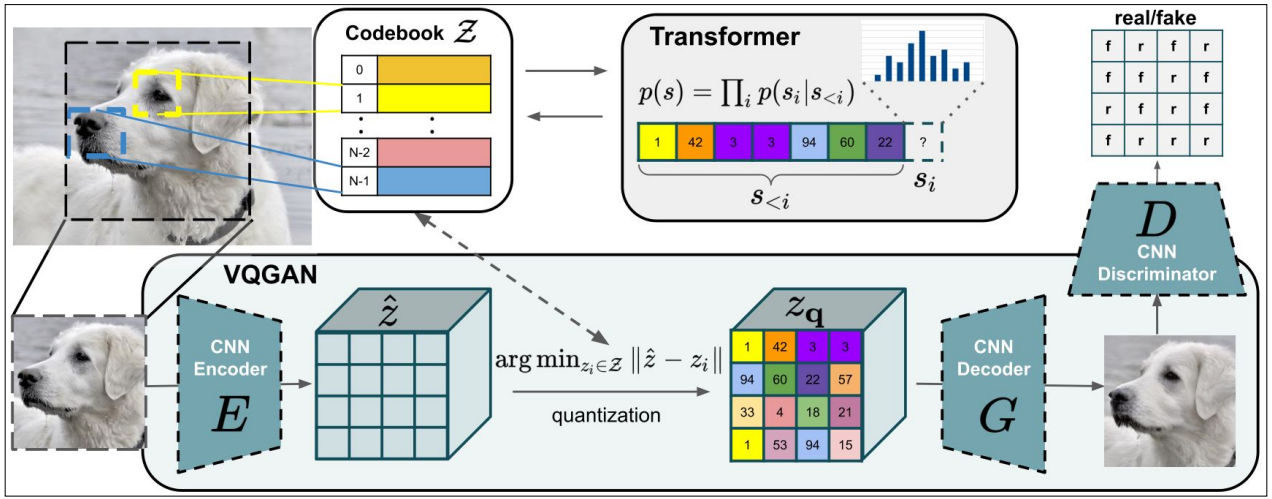
Self-Prediction: Hybrid Self-Prediction Models

VQ-VAE + AR

- Jukebox (Dhariwal et al. 2020), DALL-E (Ramesh et al. 2021)

VQ-VAE + AR + Adversarial

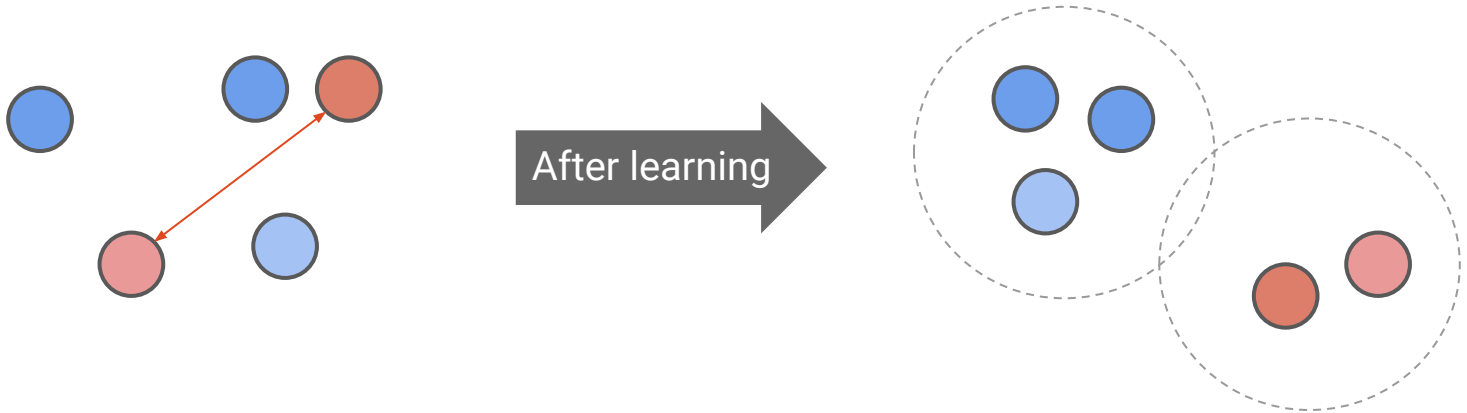
- VQGAN (Esser & Rombach et al.2021)



Methods: Contrastive Learning

Contrastive Learning

The goal of contrastive representation learning is to learn such an embedding space in which *similar* sample pairs stay *close* to each other while *dissimilar* ones are *far apart*.



Contrastive Learning

The goal of contrastive representation learning is to learn such an embedding space in which *similar* sample pairs stay *close* to each other while *dissimilar* ones are *far apart*.

1. Inter-sample classification
2. Feature clustering
3. Multiview coding

Contrastive Learning: Inter-Sample Classification

Given both similar (“positive”) and dissimilar (“negative”) candidates, to identify which ones are similar to the anchor data point is a *classification* task.

There are creative ways to construct a set of data point candidates:

1. The original input and its distorted version
2. Data that captures the same target from different views

Contrastive Learning: Inter-Sample Classification

Common loss functions:

- Contrastive loss (Chopra et al. 2005)
- Triplet loss (Schroff et al. 2015; FaceNet)
- Lifted structured loss (Song et al. 2015)
- Multi-class n-pair loss (Sohn 2016)
- Noise contrastive estimation (“NCE”; Gutmann & Hyvarinen 2010)
- InfoNCE (van den Oord, et al. 2018)
- Soft-nearest neighbors loss (Salakhutdinov & Hinton 2007, Frosst et al. 2019)

Contrastive Learning: Inter-Sample Classification

Contrastive loss (Chopra et al. 2005): Works with labelled dataset.

Encodes data into an embedding vector such that examples from the same class have similar embeddings and samples from different classes have different ones.

Given two labeled data pairs $(\mathbf{x}_i, y_i)$ and $(\mathbf{x}_j, y_j)$:

$$\mathcal{L}_{\text{cont}}(\mathbf{x}_i, \mathbf{x}_j, \theta) = \mathbb{1}[y_i = y_j] \|f_\theta(\mathbf{x}_i) - f_\theta(\mathbf{x}_j)\|_2^2 + \mathbb{1}[y_i \neq y_j] \max(0, \epsilon - \|f_\theta(\mathbf{x}_i) - f_\theta(\mathbf{x}_j)\|_2)^2$$

minimize

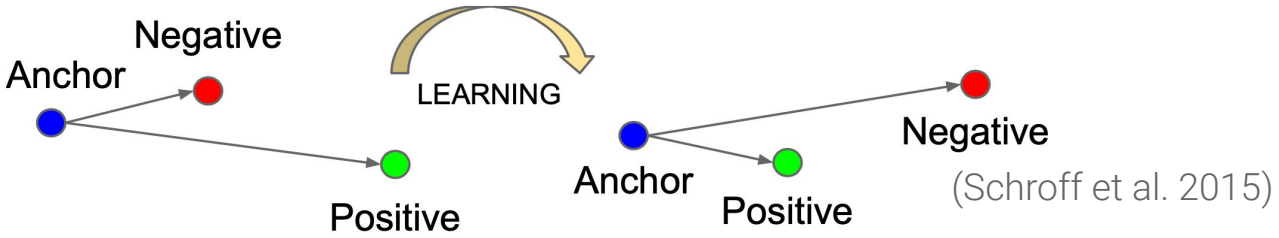
maximize

Contrastive Learning: Inter-Sample Classification

Triplet loss (Schroff et al. 2015): learns to minimize the distance between the anchor **x** and positive **x+** and maximize the distance between the anchor **x** and negative **x-** at the same time.

Given a triplet input $(\mathbf{x}, \mathbf{x}^+, \mathbf{x}^-)$,

$$\mathcal{L}_{\text{triplet}}(\mathbf{x}, \mathbf{x}^+, \mathbf{x}^-) = \sum_{\mathbf{x} \in \mathcal{X}} \max\left(0, \|f(\mathbf{x}) - f(\mathbf{x}^+)\|_2^2 - \|f(\mathbf{x}) - f(\mathbf{x}^-)\|_2^2 + \epsilon\right)$$



(Schroff et al. 2015)

Contrastive Learning: Inter-Sample Classification

N-pair loss (Sohn 2016) generalizes triplet loss to include comparison with multiple negative samples.

Given one positive and N-1 negative samples, $\{\mathbf{x}, \mathbf{x}^+, \mathbf{x}_1^-, \dots, \mathbf{x}_{N-1}^-\}$

$$\mathcal{L}_{\text{N-pair}}(\mathbf{x}, \mathbf{x}^+, \{\mathbf{x}_i^-\}_{i=1}^{N-1}) = \log\left(1 + \sum_{i=1}^{N-1} \exp(f(\mathbf{x})^\top f(\mathbf{x}_i^-) - f(\mathbf{x})^\top f(\mathbf{x}^+))\right)$$

$$= -\log \frac{\exp(f(\mathbf{x})^\top f(\mathbf{x}^+))}{\exp(f(\mathbf{x})^\top f(\mathbf{x}^+)) + \sum_{i=1}^{N-1} \exp(f(\mathbf{x})^\top f(\mathbf{x}_i^-))}$$

Contrastive Learning: Inter-Sample Classification

Lifted structured loss (Song et al. 2015): utilizes all the pairwise edges within one training batch for better computational efficiency.

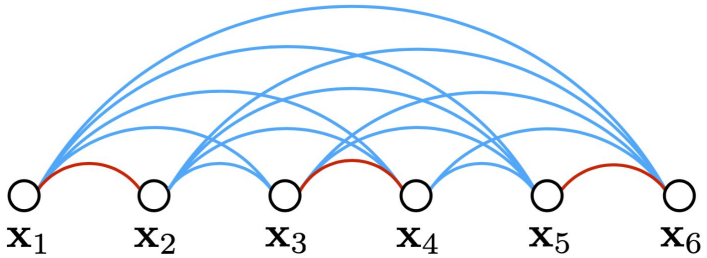
$$\mathcal{L}_{\text{struct}}^{(ij)} = D_{ij} + \log\left(\sum_{(i,k)\in\mathcal{N}} \exp(\epsilon - D_{ik}) + \sum_{(j,l)\in\mathcal{N}} \exp(\epsilon - D_{jl})\right)$$

where $D_{ij} = \|f(\mathbf{x}_i) - f(\mathbf{x}_j)\|_2$

$(i, j) \in \mathcal{P}$

$\mathcal{P}$ set of positive pairs

$\mathcal{N}$ set of negative pairs



(Song et al. 2015)

Contrastive Learning: Inter-Sample Classification

Noise Contrastive Estimation (NCE) (Gutmann & Hyvarinen 2010) runs logistic regression to tell apart the target data from noise.

Given target sample distribution p and noise distribution q,

$$\mathcal{L}_{\text{NCE}} = -\frac{1}{N}\sum_{i=1}^{N}\left[\log \sigma(\ell_\theta(\mathbf{x}_i)) + \log(1 - \sigma(\ell_\theta(\tilde{\mathbf{x}}_i)))\right]$$

just cross entropy

where logit $$\ell_\theta(\mathbf{u}) = \log \frac{p_\theta(\mathbf{u})}{q(\mathbf{u})} = \log p_\theta(\mathbf{u}) - \log q(\mathbf{u})$$

sigmoid $$\sigma(\ell) = \frac{1}{1 + \exp(-\ell)} = \frac{p_\theta}{p_\theta + q}$$

Contrastive Learning: Inter-Sample Classification

InfoNCE (van den Oord, et al. 2018): uses categorical cross-entropy loss to identify the positive sample amongst a set of unrelated noise samples.

Given a context vector c, the positive sample should be drawn from the conditional distribution p(x|c), while N−1 negative samples are drawn from the proposal distribution p(x), independent from the context c.

The probability of detecting the positive sample correctly is:

$$p(C = \mathbf{pos}|X, \mathbf{c}) = \frac{f(\mathbf{x}_{\mathbf{pos}}, \mathbf{c})}{\sum_{j=1}^{N} f(\mathbf{x}_j, \mathbf{c})} \quad \text{where the density function is} \quad f(\mathbf{x}, \mathbf{c}) \propto \frac{p(\mathbf{x}|\mathbf{c})}{p(\mathbf{x})}$$

Contrastive Learning: Inter-Sample Classification

Soft-Nearest Neighbors Loss (Frosst et al. 2019) extends the loss function to include multiple positive samples given known labels.

Given a batch of samples $\{\mathbf{x}_i, y_i)\}_{i=1}^{B}$,

$$\mathcal{L}_{\text{snn}} = -\frac{1}{B}\sum_{i=1}^{B}\log\frac{\sum_{i\neq j, y_i=y_j, j=1,\ldots,B}\exp(-f(\mathbf{x}_i,\mathbf{x}_j)/\tau)}{\sum_{i\neq k, k=1,\ldots,B}\exp(-f(\mathbf{x}_i,\mathbf{x}_k)/\tau)}$$

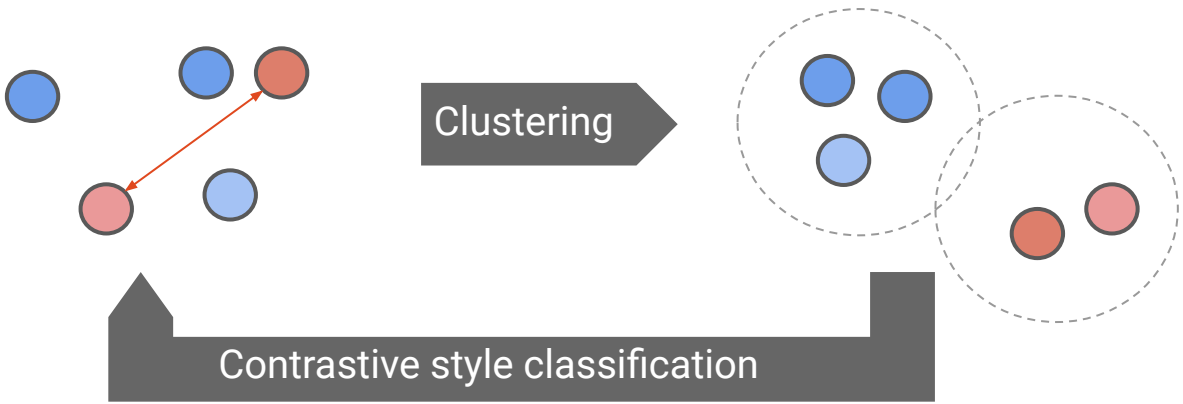
temperature term

Contrastive Learning: Feature Clustering

Find similar data samples by clustering them with learned features.

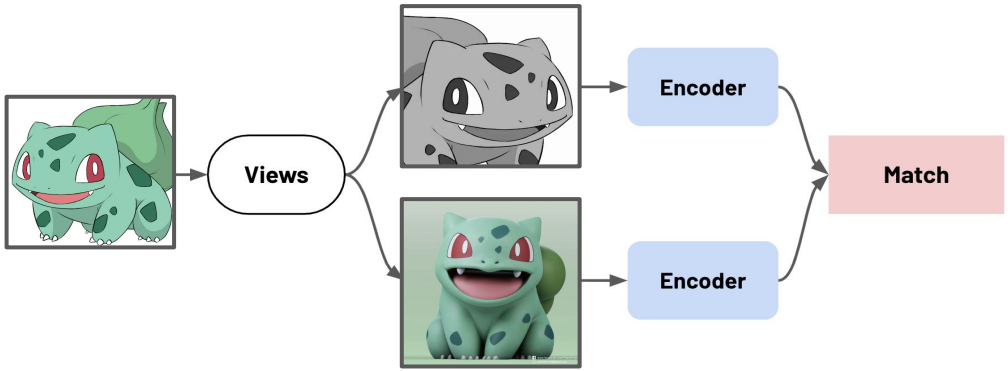
Core idea: Use clustering algorithms to assign pseudo labels to samples such that we can run intra-sample contrastive learning.

Examples: DeepCluster (Caron et al 2018); InterCLR (Xie et al 2021)



Contrastive Learning: Multiview Coding

Apply the InfoNCE objective to two or more different views of input data



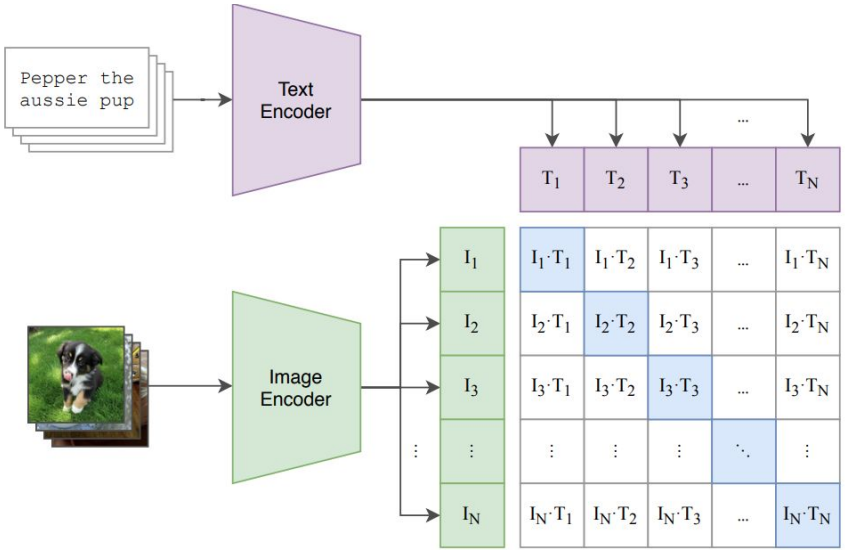
Became a mainstream contrastive learning method:

- AMDIM (Bachman et al. 2019)
- Contrastive multiview coding (CMC; Tian et al. 2019)
- And many, many more!

Contrastive Learning Between Modalities

“Views” can be from paired inputs from two or more modalities

- CLIP (Radford et al. 2021), ALIGN (Jia et al. 2021): enables zero-shot classification, cross-modal retrieval, guided image generation.
- CodeSearchNet (Husain et al 2019): contrast learning between text and code.



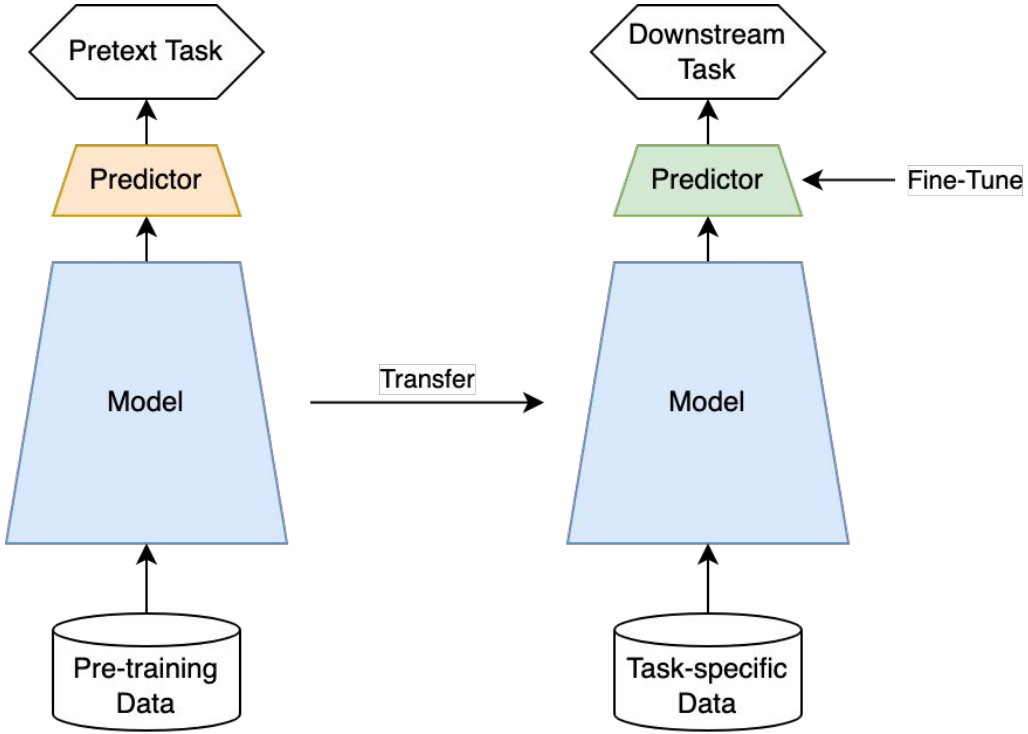
Pretext Tasks

- Vision
- Video
- Audio
- Multimodal
- Language

Recap: Pretext Tasks

Step 1: Pre-train a model for a pretext task

Step 2: Transfer to applications



Pretext Tasks: Taxonomy

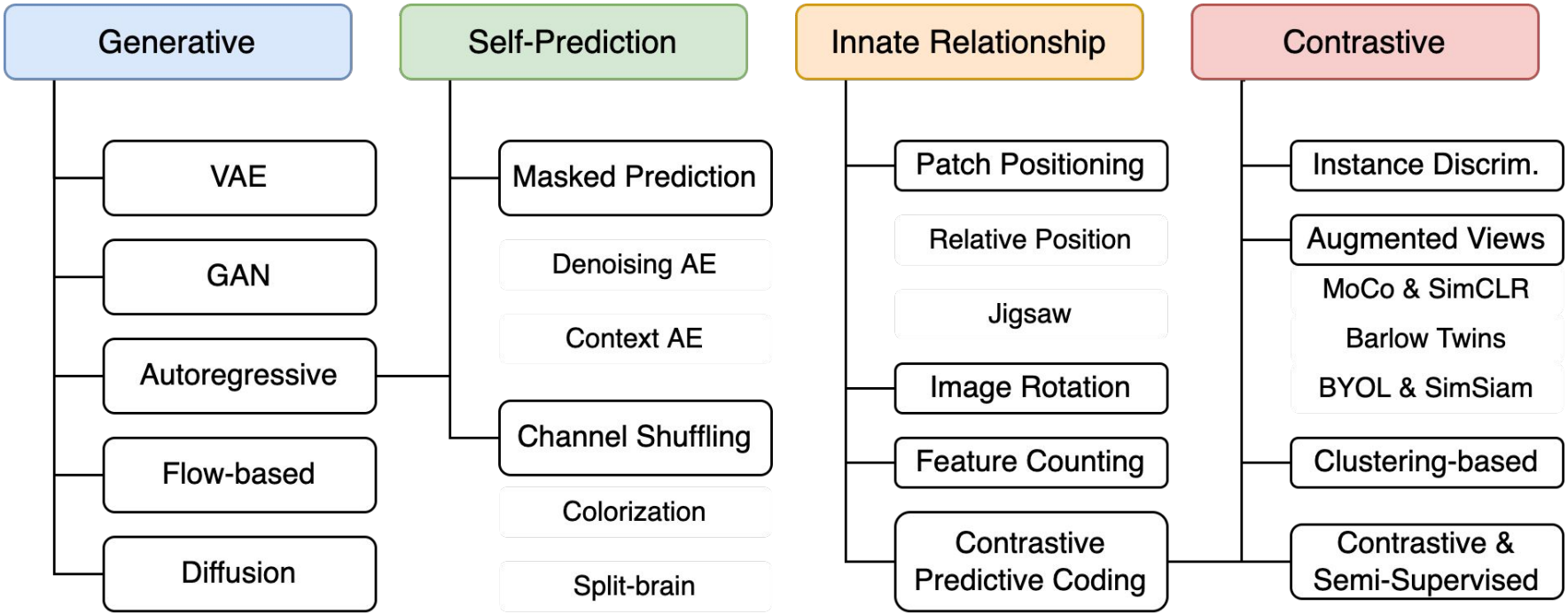
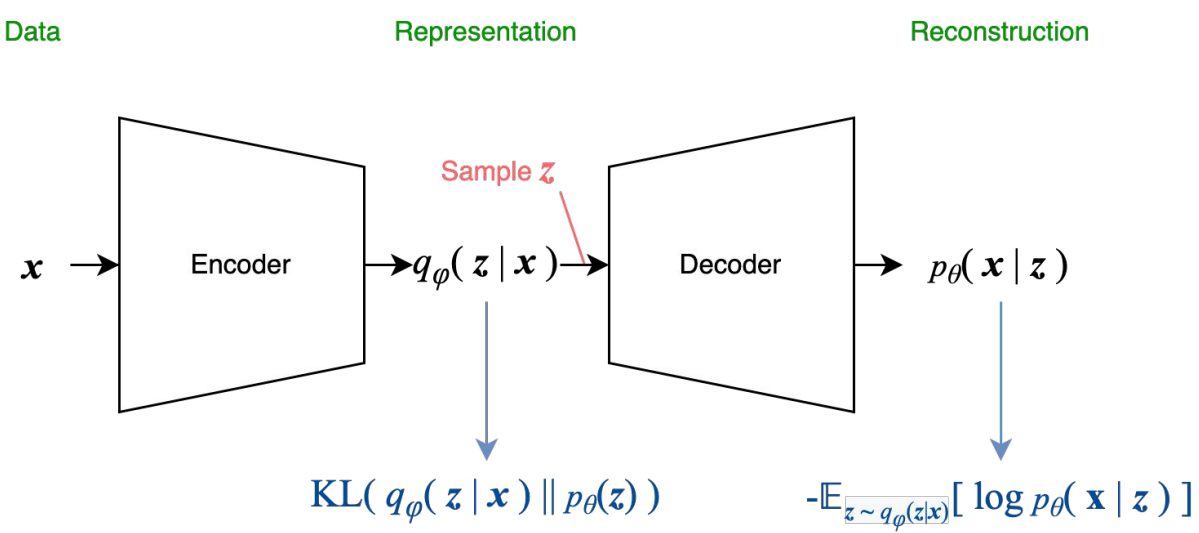
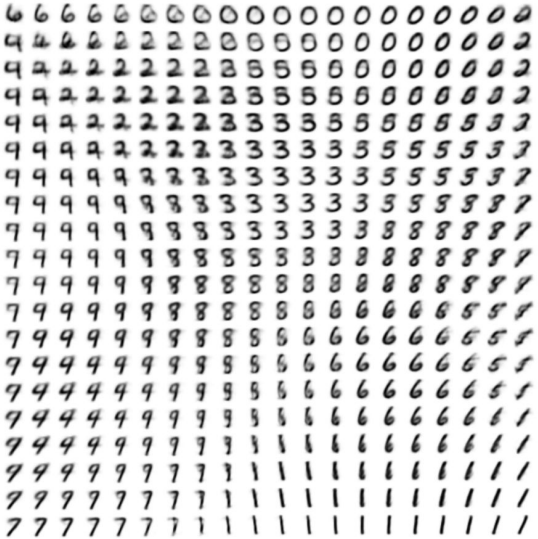


Image Pretext Tasks: Variational Autoencoders

Auto-Encoding Variational Bayes (Kingma et al. 2014)



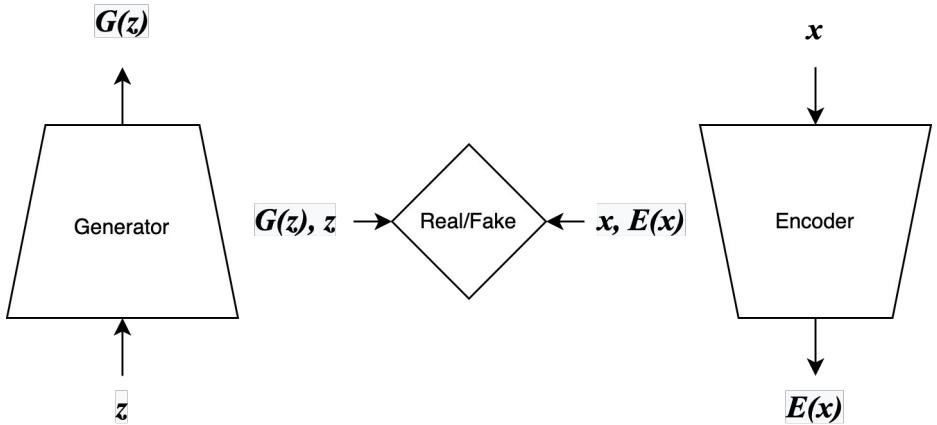


(Kingma et al. 2014)

Image Pretext Tasks: Generative Adversarial Networks

Jointly train an encoder, additional to the usual GAN (Goodfellow et al. 2014):

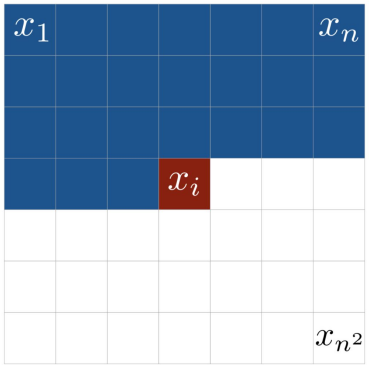
- Bidirectional GAN (BiGAN; Donahue et al. 2016)
- Adversarially Learned Inference (ALI; Dumoulin et al. 2016)



GAN inversion: learning encoder post-hoc and/or optimizing for given image

Vision Pretext Tasks: Autoregressive Image Generation

- Neural autoregressive density estimation (NADE; Larochelle et al. 2011)
- PixelRNN, PixelCNN (Oord et al. 2016)
- Image GPT (Chen et al. 2020)



Raster scan order

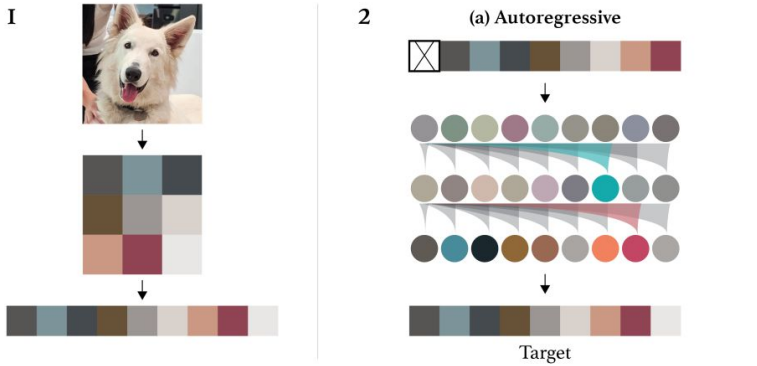
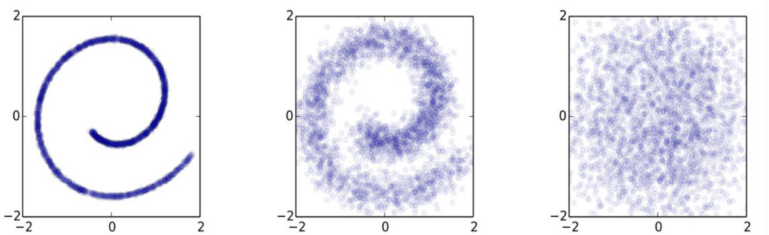


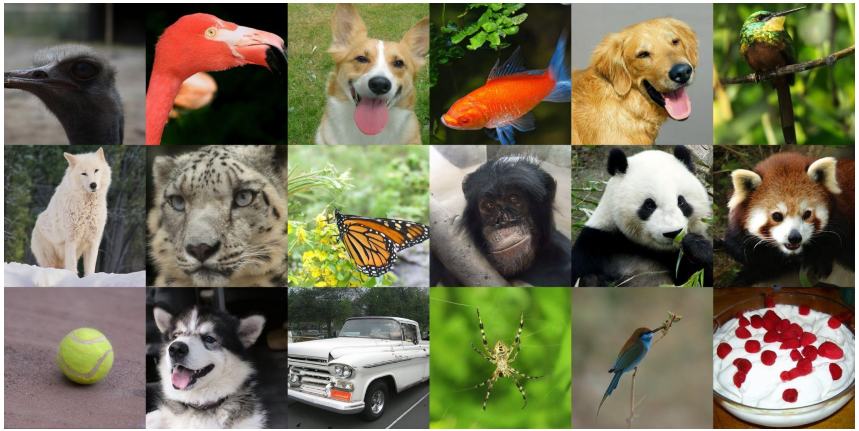
Image GPT (Chen et al. 2020)

Vision Pretext Tasks: ~~Autoregressive~~ Image Generation

Diffusion modeling: Follows a Markov chain of diffusion steps to slowly add random noise to data and then learn to reverse the diffusion process to construct desired data samples from the noise. (Sohl-Dickstein et al 2015; Yang & Ermon 2019; Ho et al. 2020; Dhariwal & Nichol 2021)



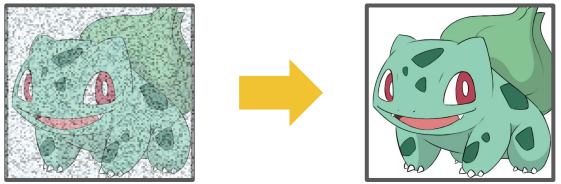
Diffusion modeling



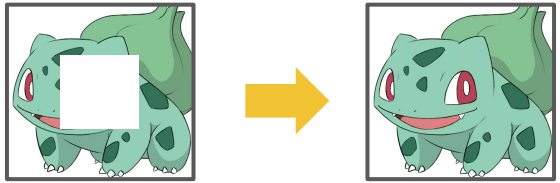
(Dhariwal & Nichol 2021)

Vision Pretext Tasks: Masked Prediction

- **Denoising autoencoder** (Vincent et al. 2008)
  - Add noise = Randomly mask some pixels
  - Only reconstruction loss

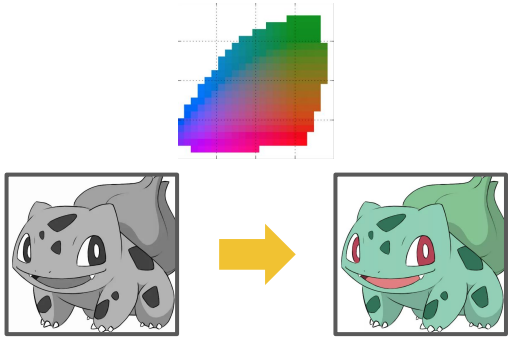


- **Context autoencoder** (Pathak et al. 2016)
  - Mask a random region in the image
  - Reconstruction loss + adversarial loss

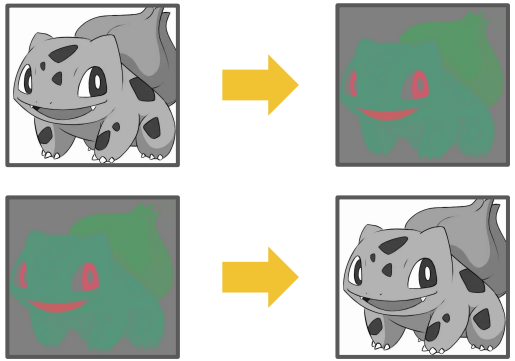


Vision Pretext Tasks: Colorization and More

- **Colorization** (Zhang et al. 2016)
  - Predict the binned CIE Lab color space given a grayscale image.

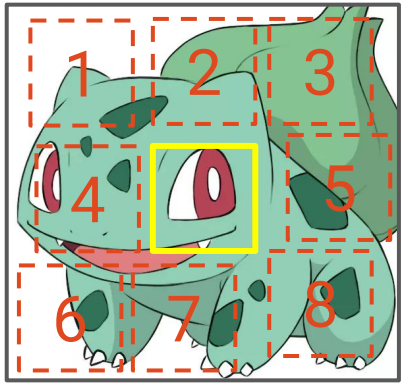


- **Split-brain autoencoder** (Zhang et al. 2017)
  - Predict a subset of color channels from the rest of channels.
  - Channels: luminosity, color, depth, etc.

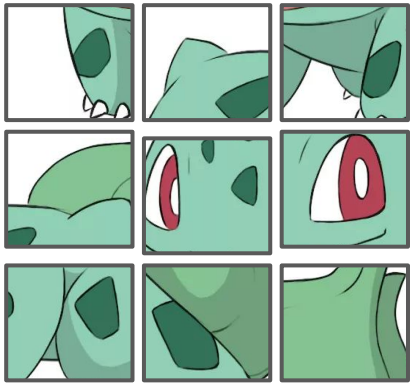


Vision Pretext Tasks: Innate Relationship Prediction

- Learn the relationship among image patches:
  - Predict **relative positions** between patches (Doersch et al 2015)
  - **Jigsaw puzzle** using patches (Noroozi & Favaro 2016)



Given a patch, predict which one of 8 neighboring locations another patch is in



Output a probability vector per patch index out of a predefined set of permutations

Vision Pretext Tasks: Innate Relationship Prediction

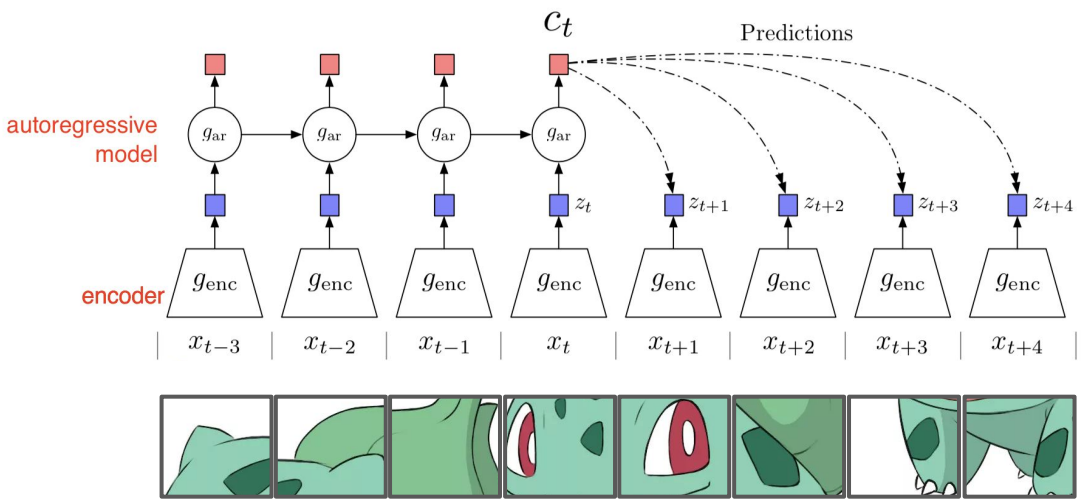
- RotNet: predict **which rotation** is applied (Gidaris et al. 2018)
  - Rotation does not alter the semantic content of an image.

- Representation Learning by Learning to Count (Noroozi et al. 2017)
  - Counting features across patches without labels, using equivariance of counts

Contrastive Predictive Coding and InfoNCE

Contrastive Predictive Coding (CPC) (van den Oord et al. 2018)

- classify the “future” representation amongst a set of unrelated “negative” samples



(van den Oord et al. 2018)

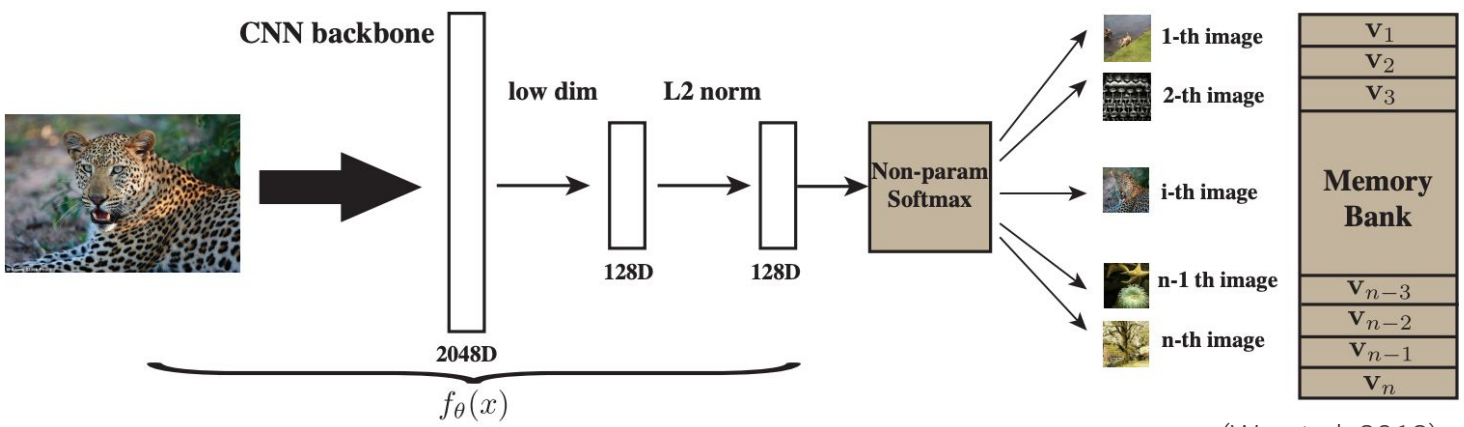
The InfoNCE loss

$$\mathcal{L}_{\text{CPC}} = -\mathbb{E}_X \left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{i=1}^{N} f_k(x_i, c_t)} \right]$$

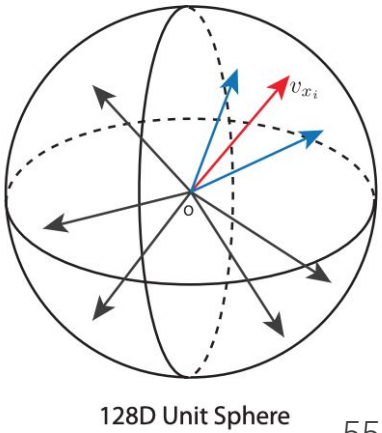
A density function to preserve the mutual information between x_{t+k} and c_t

Vision Pretext Tasks: Inter-Sample Classification

- **Exemplar CNN** (Dosovitskiy et al. 2014)
- **Instance-level discrimination** (InstDisc; Wu et al. 2018)
  - Each instance is *a distinct class of its own*: # classes = # training samples
  - Non-parametric softmax that compares features: $\frac{\exp\left(\mathbf{v}_i^T\mathbf{v}/\tau\right)}{\sum_{j=1}^{n}\exp\left(\mathbf{v}_j^T\mathbf{v}/\tau\right)}$
  - Memory bank for storing representations of past samples, $V = \{\mathbf{v}_i\}$

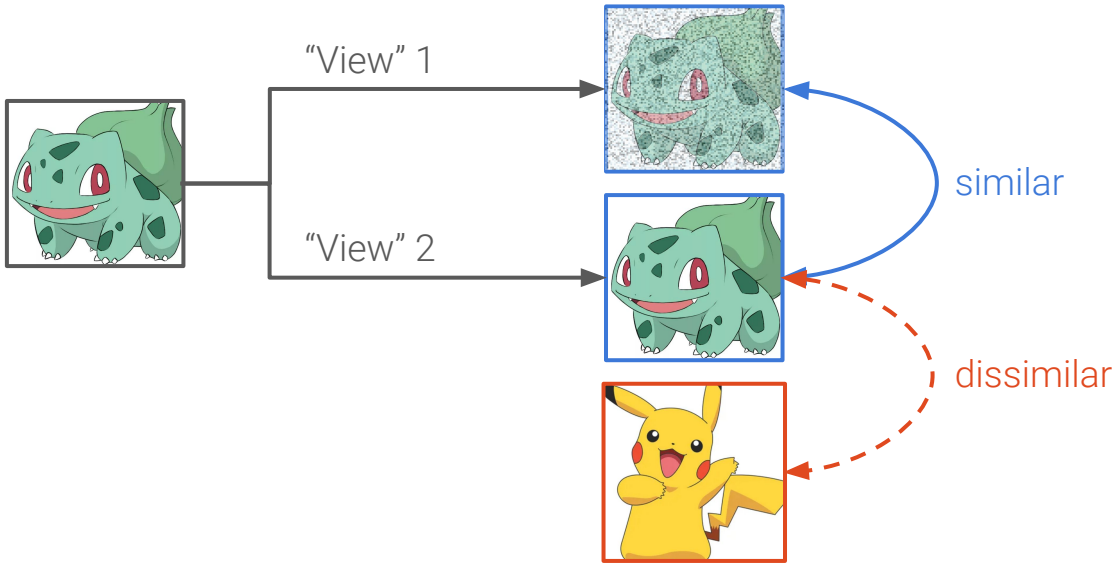


(Wu et al. 2018)



Vision Pretext Tasks: Contrastive Learning

The common approach is to make multiple views (e.g. data augmentation) to one image and consider the image and its distorted version as similar pairs, while different images are treated dissimilar.



Vision Pretext Tasks: Data Augmentation and Multiple Views

Augmented Multiscale Deep InfoMax
(**AMDIM**; Bachman et al. 2019)

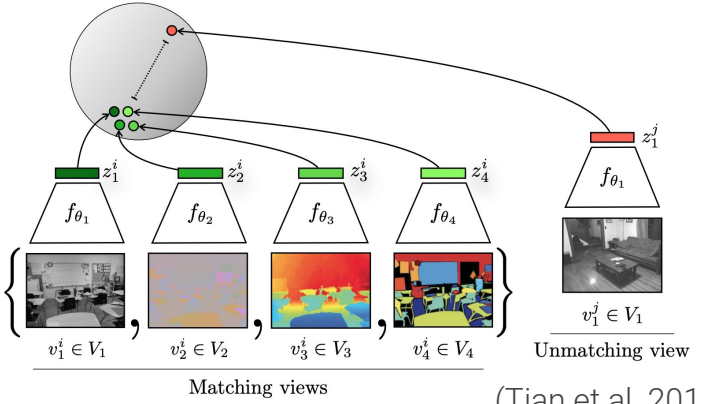
- Views from different **augmentations**

Contrastive Multiview Coding
(**CMC**; Tian et al. 2019)

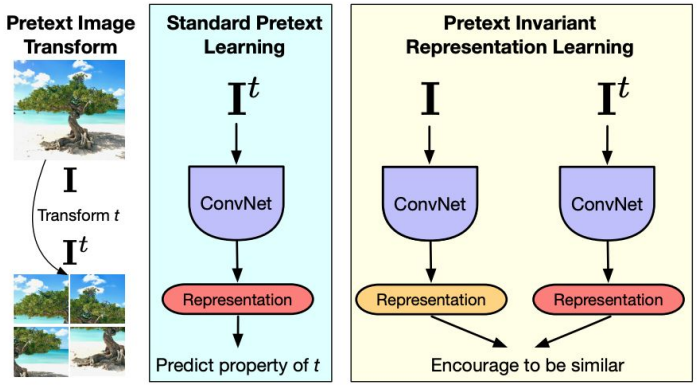
- Multiple views from different **channels**

Pretext-Invariant Representation Learning
(**PIRL**; Misra et al. 2019)

- Jigsaw transformation



(Tian et al. 2019)

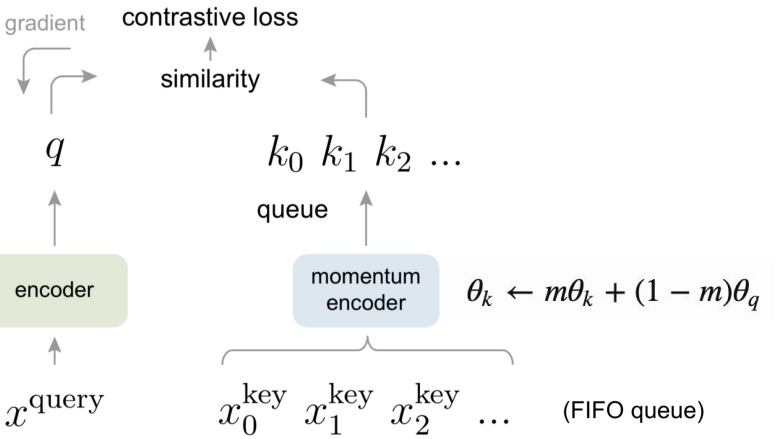


(Misra et al. 2019)

Vision Pretext Tasks: Inter-Sample Classification

MoCo (Momentum Contrast; He et al. 2019)

- Memory bank is a FIFO queue now
- The target features are encoded using a momentum encoder
- MoCo v2 (Chen et al. 2020): MLP projection head & stronger data augmentation
- MoCo v3 (Chen et al. 2021): Vision Transformer, in-batch negatives



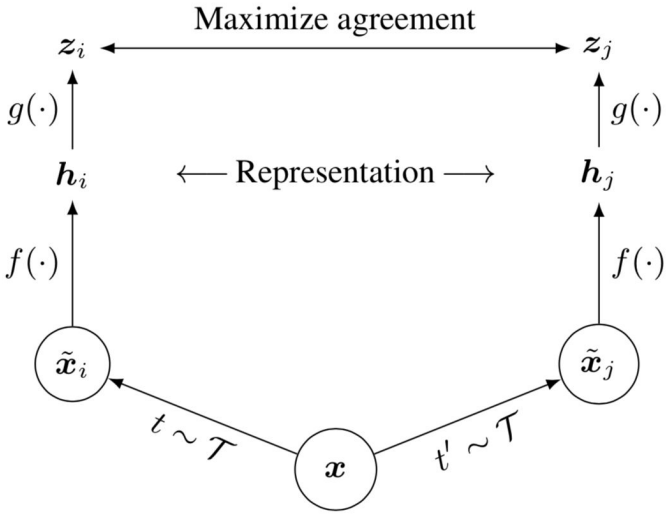
(He et al. 2019)

(In-batch negatives in MoCo V3)

Vision Pretext Tasks: Inter-Sample Classification

SimCLR (Simple framework for Contrastive Learning of visual Representations)

- Contrastive learning loss
- SimCLR (Chen et al. 2020 Feb)
  - f(.) - base encoder
  - g(.) - projection head layer
  - In-batch negative samples
- SimCLRv2 (Chen et al. 2020 Jun)
  - Larger ResNet models
  - Deeper g(.)
  - Memory bank



$$\mathbf{h}_i = f(\tilde{\mathbf{x}}_i), \quad \mathbf{h}_j = f(\tilde{\mathbf{x}}_j) \qquad \mathbf{z}_i = g(\mathbf{h}_i), \quad \mathbf{z}_j = g(\mathbf{h}_j)$$

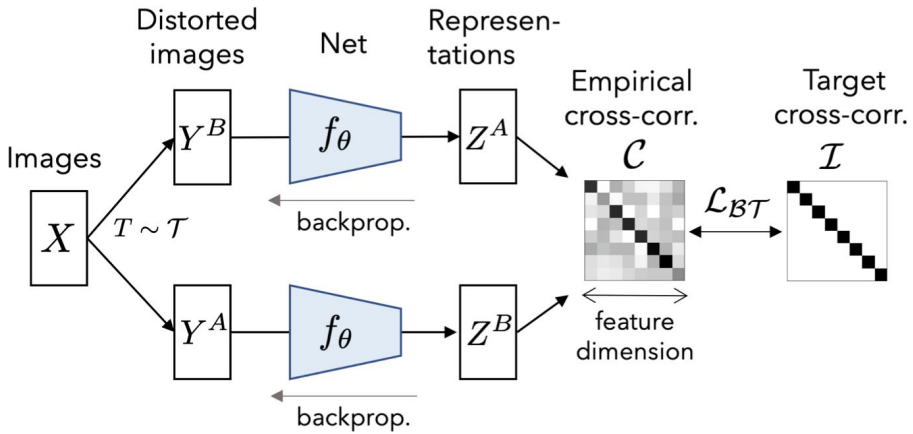
$$\mathcal{L}_{\text{SimCLR}}^{(i,j)} = -\log \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau)}$$

(Chen et al. 2020)

Vision Pretext Tasks: Inter-Sample Classification

Barlow Twins (Zbontar et al. 2021)

- Learn to make the cross-correlation matrix between two output features for two distorted version of the same sample close to the identity.



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

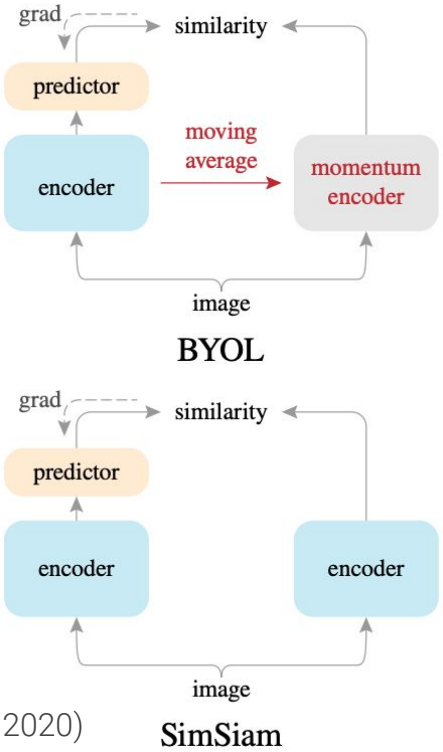
$$\text{where } C_{ij} = \frac{\sum_b \mathbf{z}^A_{b,i} \mathbf{z}^B_{b,j}}{\sqrt{\sum_b (\mathbf{z}^A_{b,i})^2} \sqrt{\sum_b (\mathbf{z}^B_{b,j})^2}}$$

(Zbontar et al. 2021)

Vision Pretext Tasks: Non-Contrastive Siamese Networks

Learn similarity representations for different augmented views of the same sample, but no contrastive component involving negative samples.

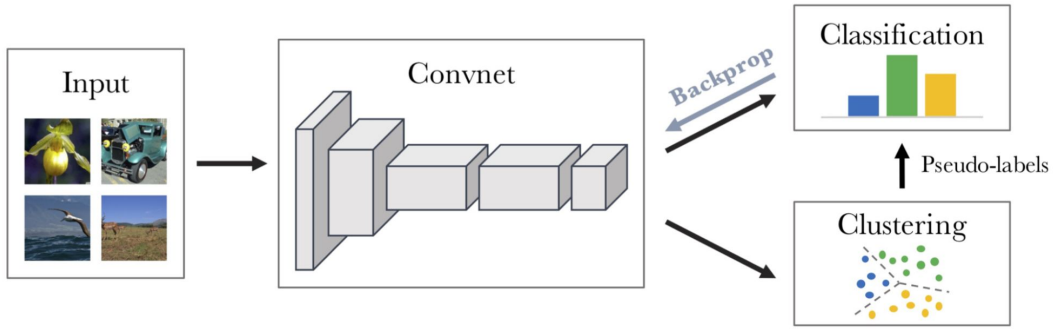
- Minimize L2 distance between online and target features
- **Bootstrap Your Own Latent** (BYOL; Grill et al. 2020)
  - Momentum-encoded features as the target
- **SimSiam** (Chen & He 2020)
  - No momentum encoder
  - Large batch size unnecessary
- BatchNorm seems to be playing an important role



(Chen & He 2020)

Vision Pretext Tasks: Feature Clustering with K-Means

- **DeepCluster** (Caron et al. 2018): Iteratively clusters features via k-means and uses cluster assignments as pseudo labels to provide supervised signals.
- **Online DeepCluster** (Zhan et al. 2020): Performs clustering and network update simultaneously rather than alternatingly.

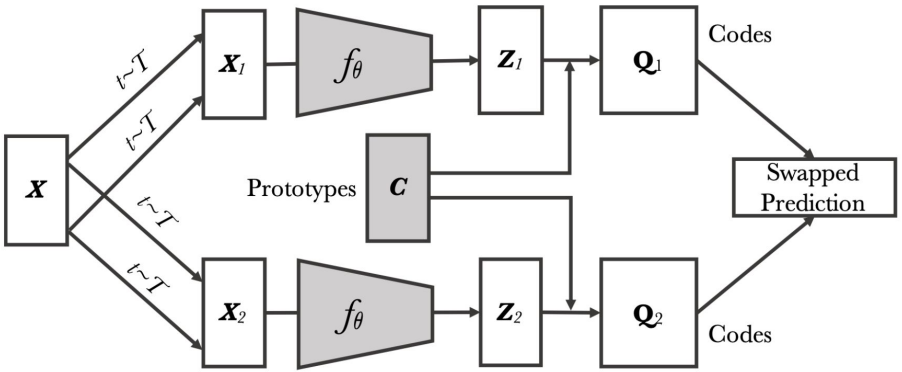


(Zhan et al. 2020)

- **Prototypical Cluster Learning** (PCL; Li et al. 2020): Online EM for clustering, combined with InfoNCE for smoothness

Vision Pretext Tasks: Feature Clustering with Sinkhorn-Knopp

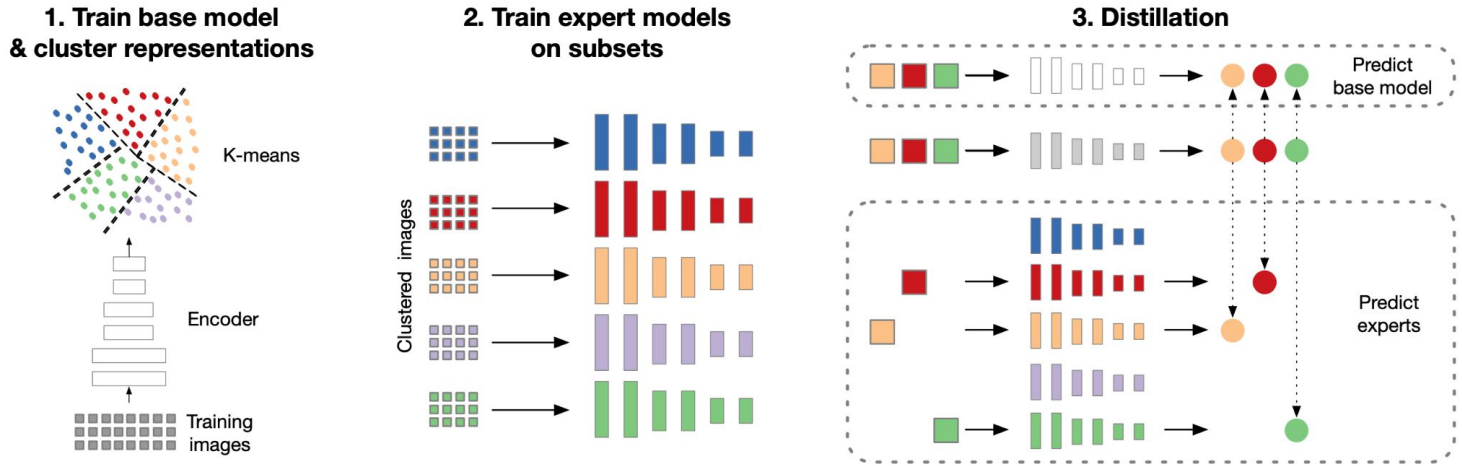
- **SeLa** (Self-Labelling; Asano et al. 2020)
- **SwAV** (Swapping Assignments between multiple Views; Caron et al. 2020)
    - Implicit clustering via a learned *prototype code* (“anchor clusters”).
    - Predict cluster assignment in the other column.



(Caron et al. 2020)

Vision Pretext Tasks: Feature Clustering to Improve SSL

- **InterCLR** (Xie et al. 2020): Inter-sample contrastive pairs are constructed according to pseudo labels obtained by clustering.
- **Divide and Contrast** (Tian et al. 2021): Train expert models on the clustered datasets and then distill the experts into a single model.

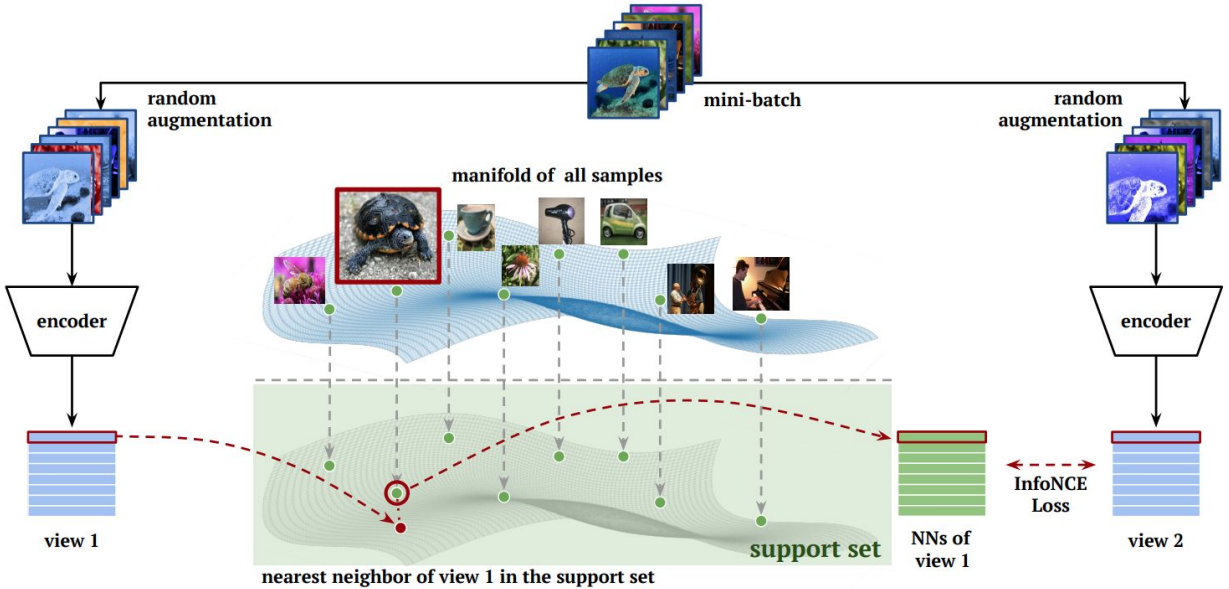


(Tian et al. 2021)

Vision Pretext Tasks: Nearest-Neighbor

NNCLR (Dwibedi et al. 2021)

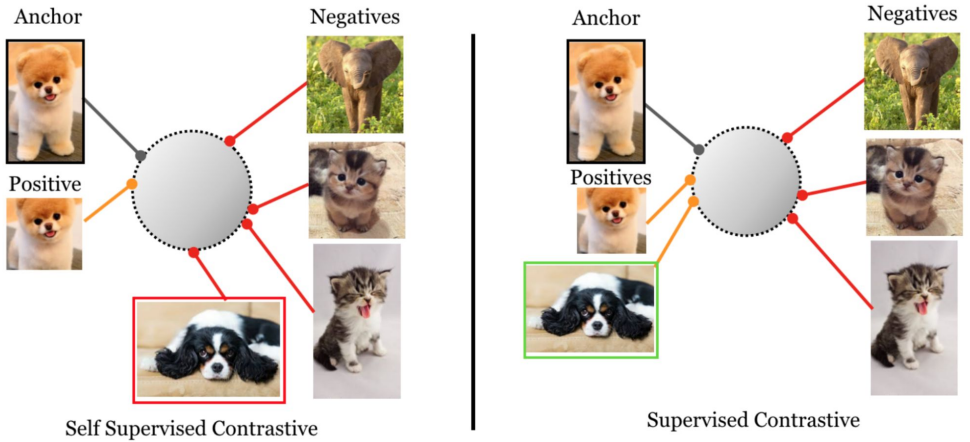
- Contrast with the nearest neighbors in the embedding space
- Allows for lighter data augmentation for views



(Dwibedi et al. 2021)

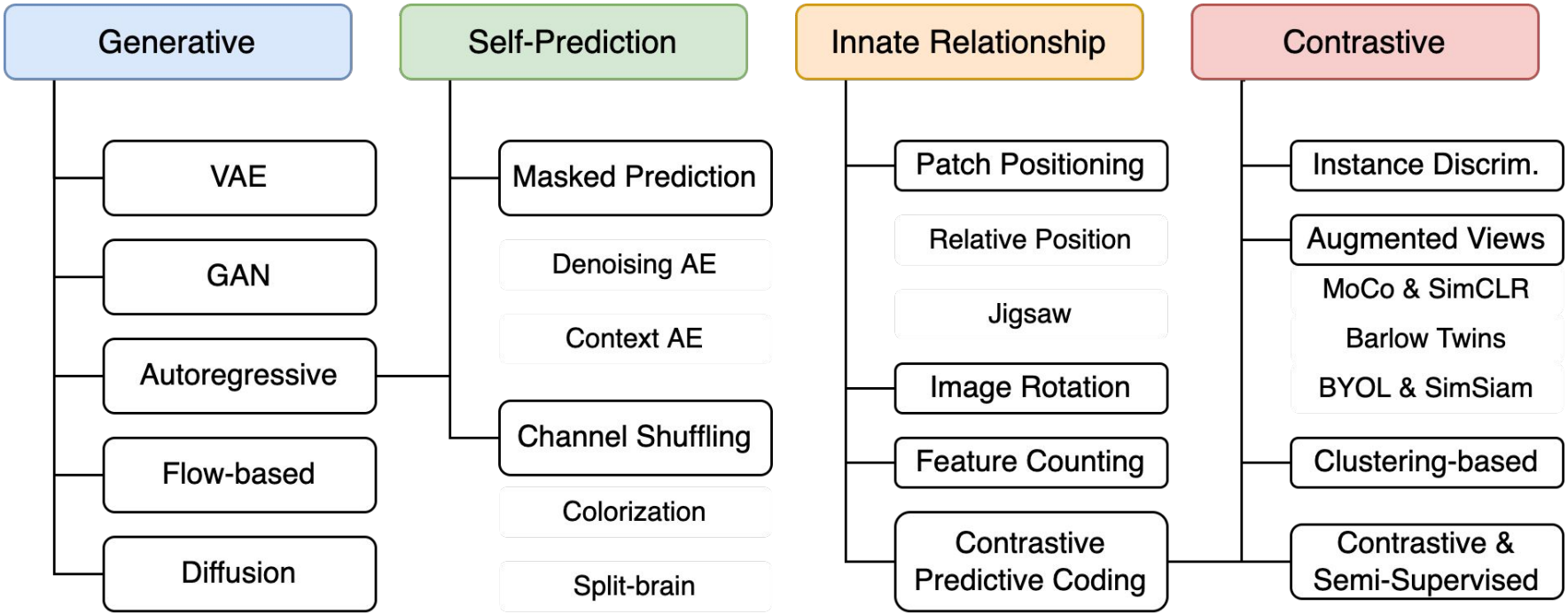
Vision Pretext Tasks: Combining with Supervised Loss

- Combine supervised loss + self-supervised learning
  - Self-supervised semi-supervised learning (**S4L**; Zhai et al 2019)
  - Unsupervised data augmentation (**UDA**; Xie et al 2019)
- Use known labels for contrastive learning
  - **Supervised Contrastive Loss** (SupCon; Khosla et al. 2021)



(Khosla et al. 2021)

Pretext Tasks: Taxonomy



Video Pretext Tasks: Innate Relationship Prediction

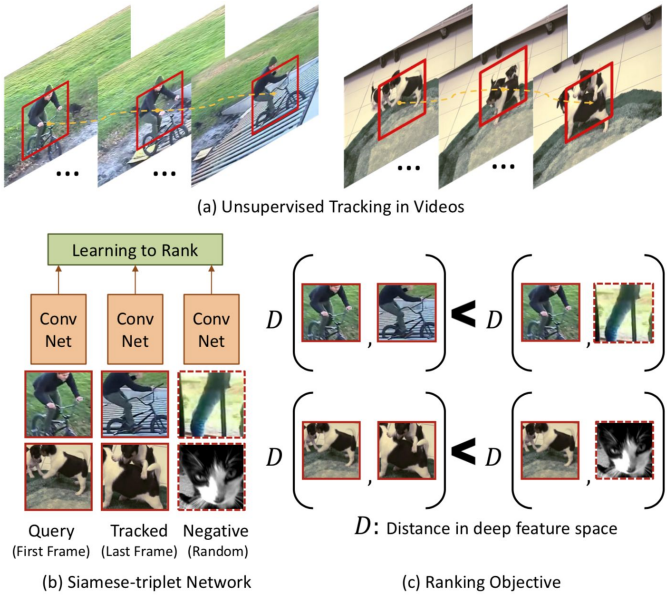
Most image pretext tasks can be applied to videos. However, with an additional time dimension, much more information about the video shot configuration or the physical world can be extracted from videos.

- Predicting object movements
- 3D motion of camera

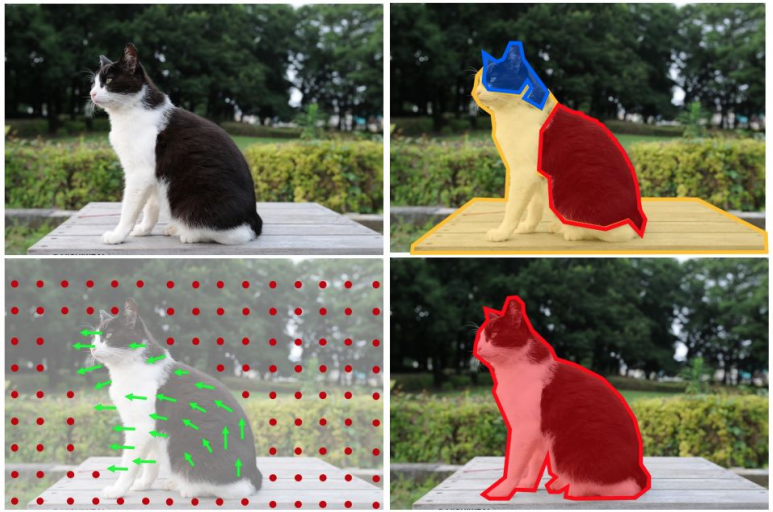
Video Pretext Tasks: Optical Flow

Tracking object movement tracking in time

- Tracking movement of image patches (Wang & Gupta 2016)

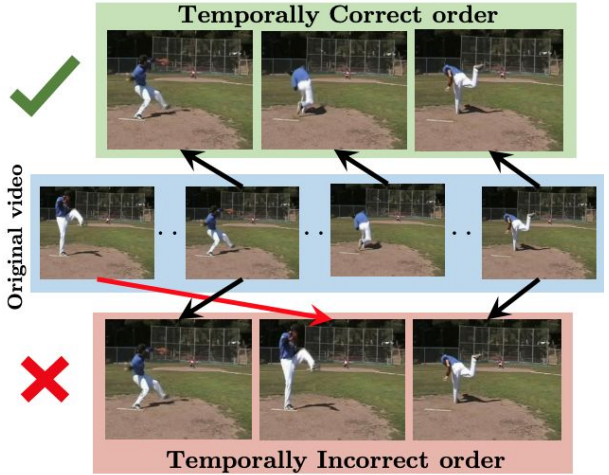


- Segmenting based on motion (Pathak et al. 2017)

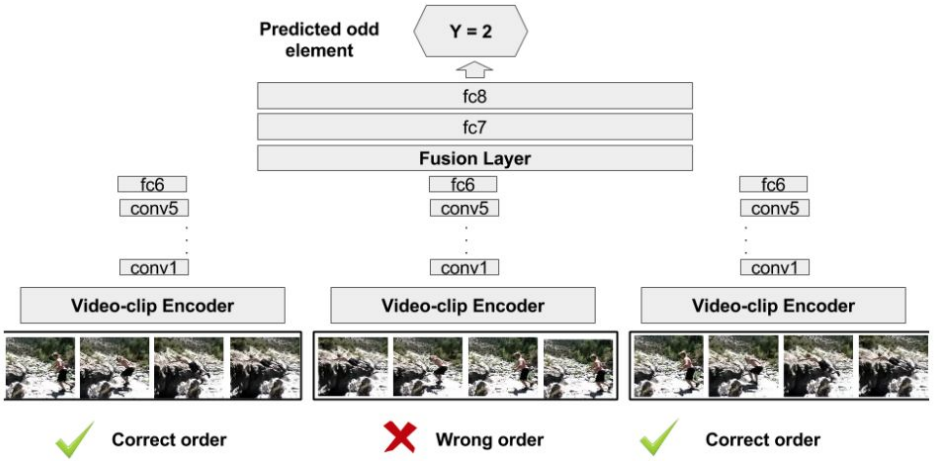


Video Pretext Tasks: Sequence Ordering

- Temporal order verification (Misra et al. 2016, Fernando et al. 2017)



(Misra et al. 2016)



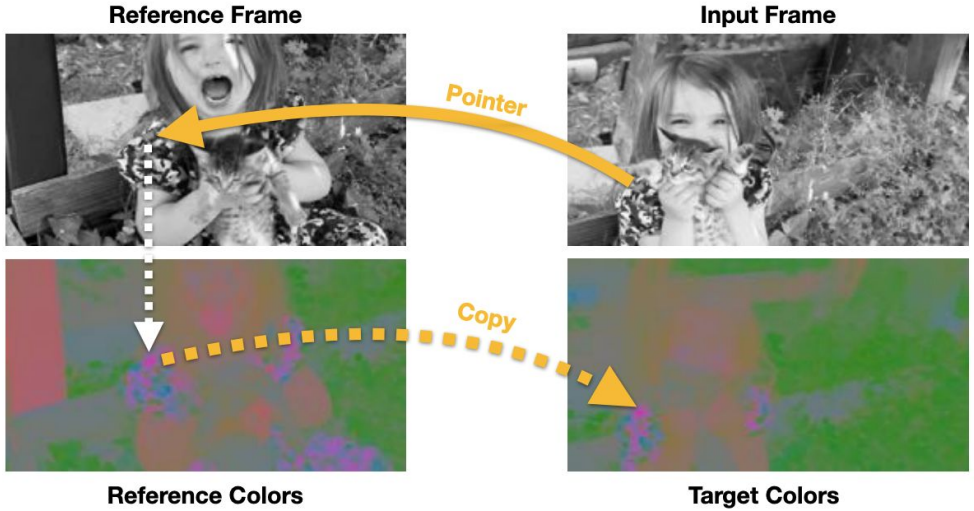
(Fernando et al. 2017)

- Predict *the arrow of time*, forward or backward (Wei et al. 2018)

Video Pretext Tasks: Colorization

Tracking emerges by colorizing videos (Vondrick et al. 2018)

- Copy colors from a reference frame to another target frame in grayscale by leveraging the natural temporal coherence of colors across video frames.

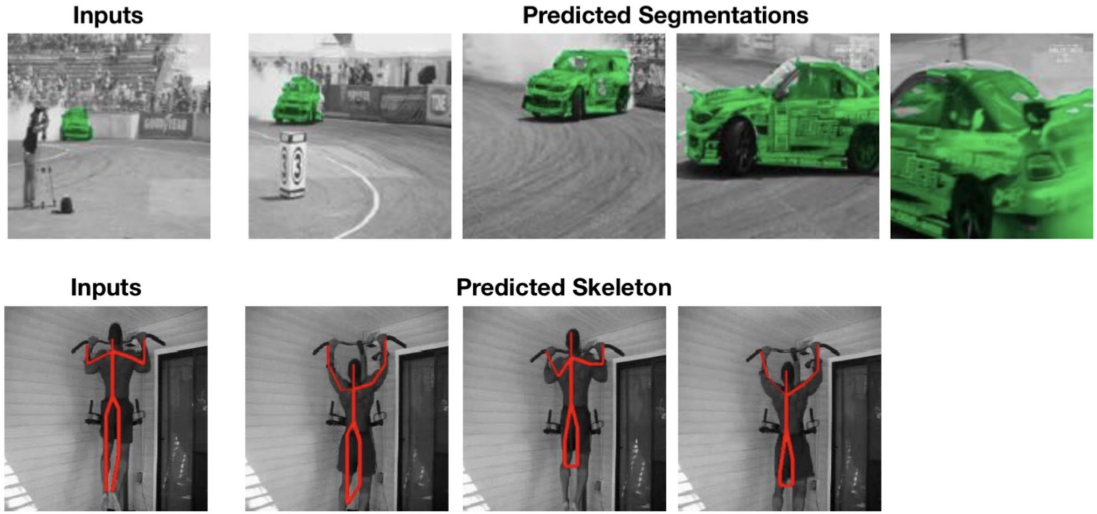


(Vondrick et al. 2018)

Video Pretext Tasks: Colorization

Tracking emerges by colorizing videos (Vondrick et al. 2018)

- Used for video segmentation or human pose estimation without fine-tuning!



(Vondrick et al. 2018)

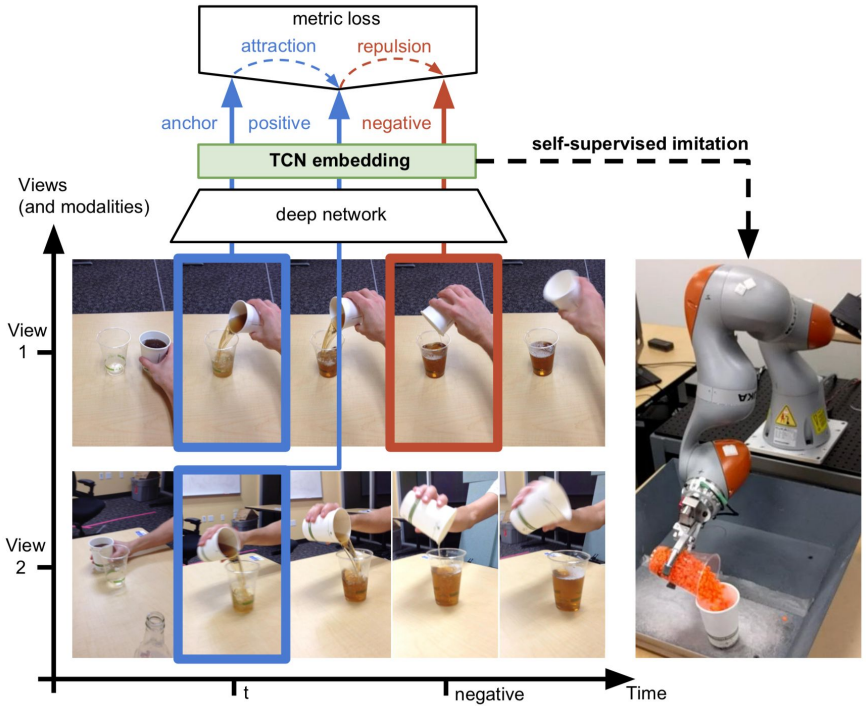
Video Pretext Tasks: Contrastive Multi-View Learning

TCN (Sermanet et al. 2017)

- Use triplet loss
- Different viewpoints at the same timestep of the same scene should share the same embedding, while embedding should vary in time, even of the same camera viewpoint.

Multi-frame TCN (Dwibedi et al. 2019)

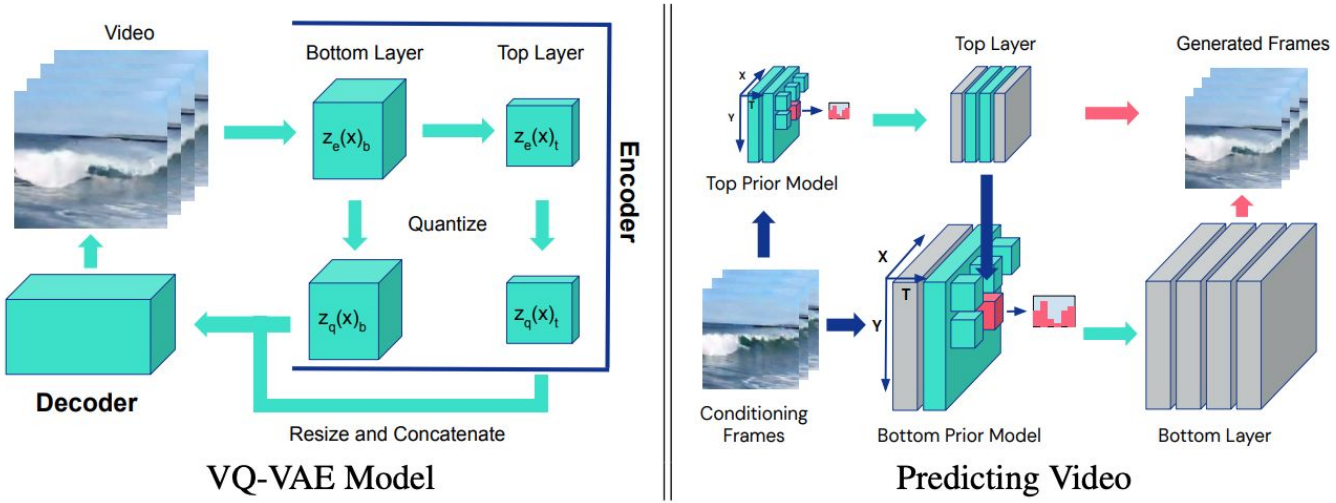
- Use n-pairs loss
- Multiple frames are aggregated into one embedding.



(Sermanet et al. 2017)

Video Pretext Task: Autoregressive Generation

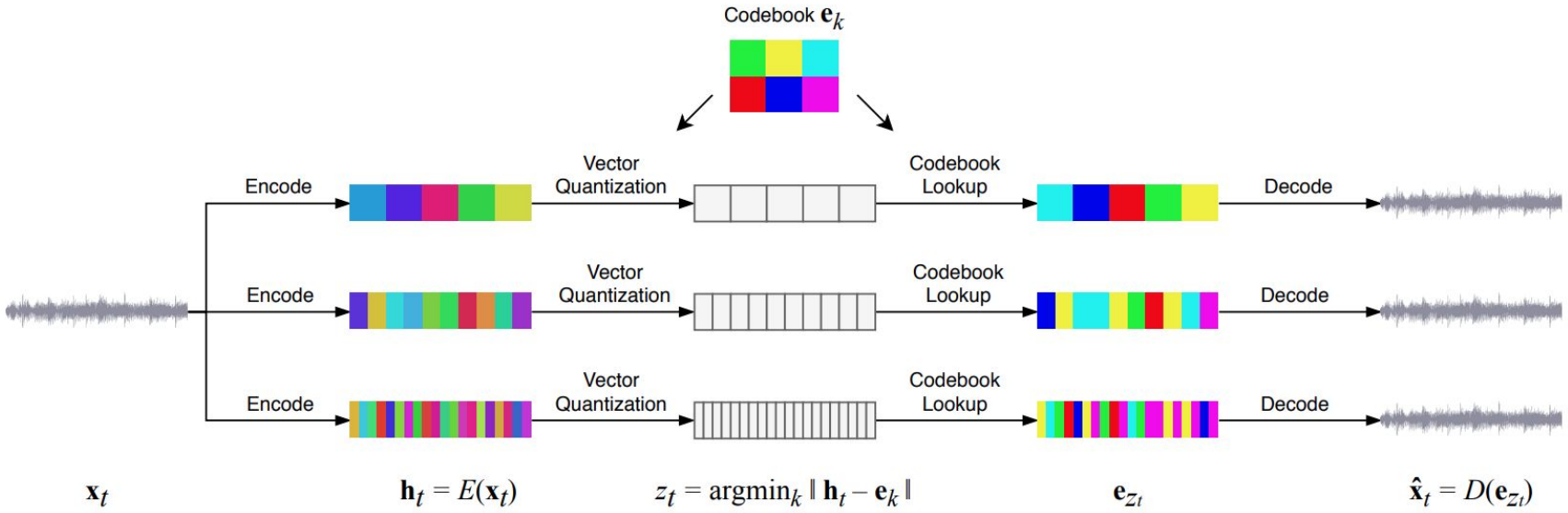
- Predicting videos with VQ-VAE (Walker et al. 2021)
- VideoGPT: Video generation using VQ-VAE and Transformers (Yan et al. 2021)



(Walker et al. 2021)

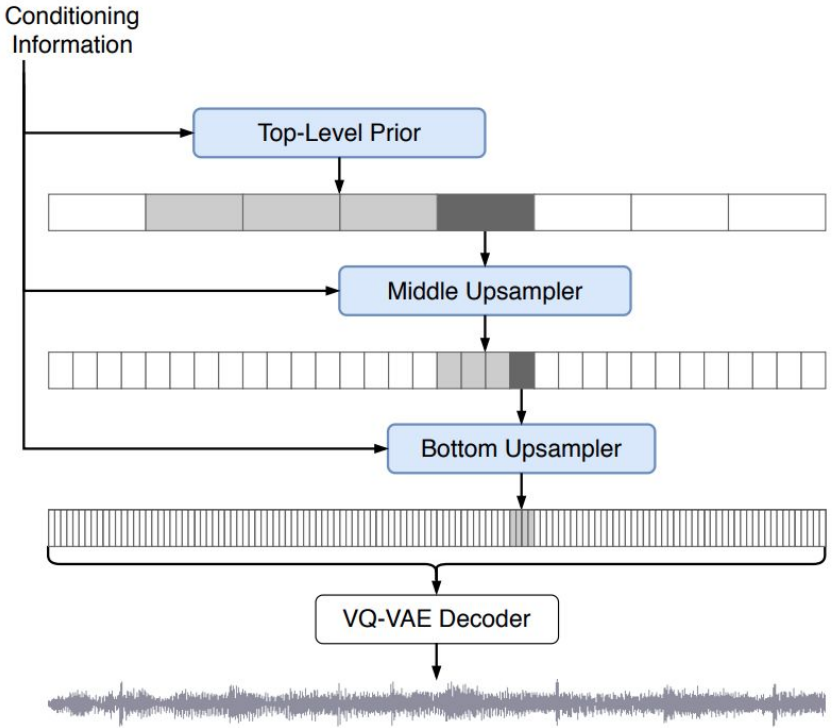
Audio Pretext Task: Autoregressive Modeling

Jukebox (Dhariwal et al. 2020)



Audio Pretext Task: Autoregressive Modeling

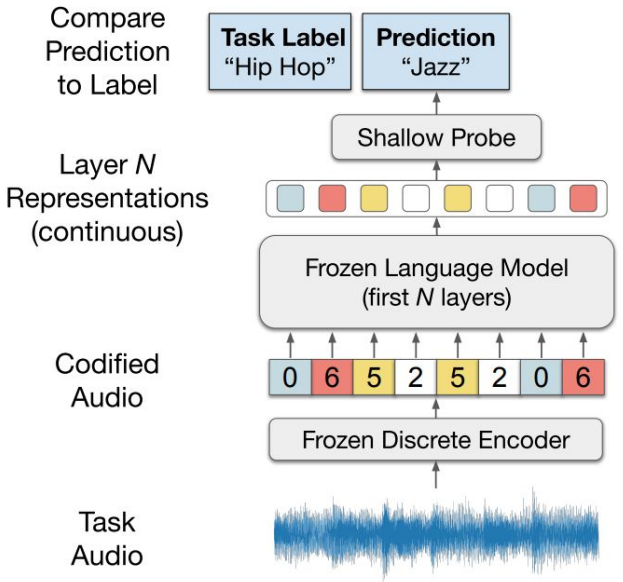
Jukebox (Dhariwal et al. 2020)



Audio Pretext Task: Autoregressive Modeling

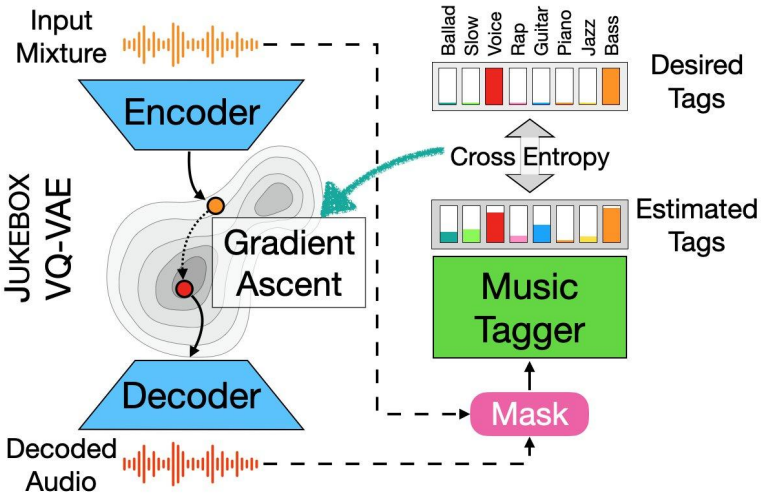
CALM (Castellon et al. 2021)

- Jukebox representation for MIR tasks



TagBox (Manilow et al. 2021)

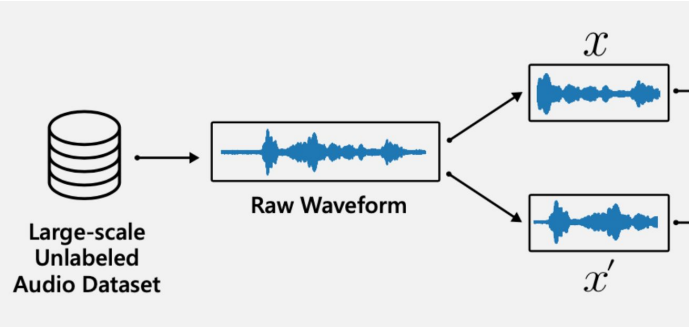
- Source separation by steering Jukebox' latent space

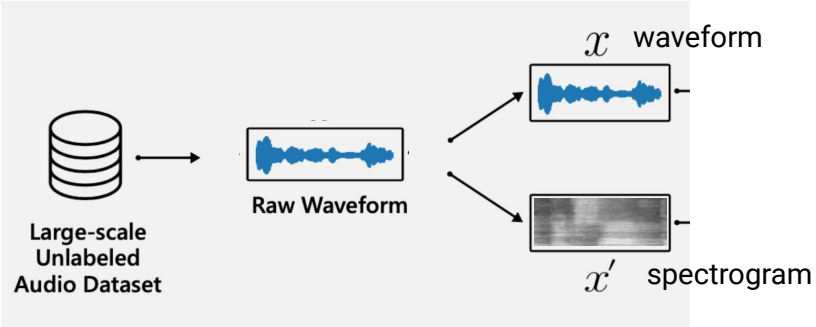


Audio Pretext Tasks: Contrastive Learning

COLA (Saeed et al. 2021) assigns high similarity between audio clips extracted from the same recording and low similarity to clips from different recordings.

Multi-Format audio contrastive learning (Wang & van den Oord 2021) assigns high similarity between the raw audio format and the corresponding spectral representation.

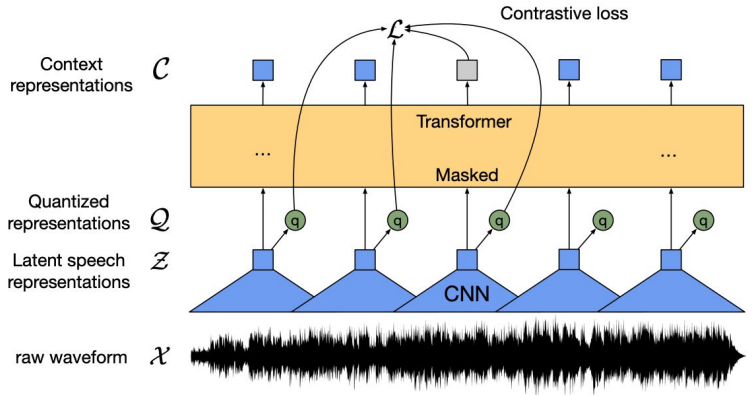




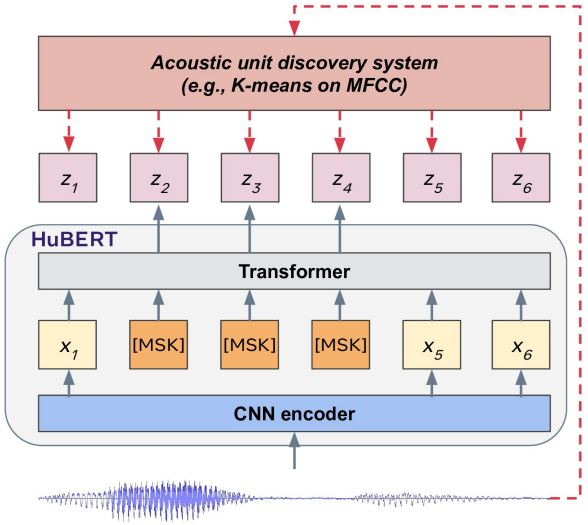
(Saeed et al. 2021)

Audio Pretext Task: Masked Language Modeling for ASR

Wav2Vec 2.0 (Baevski et al. 2020)



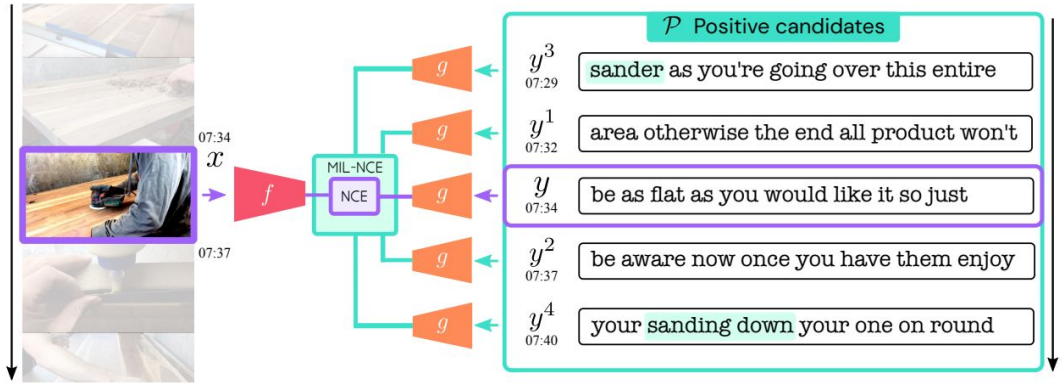
HuBERT (Hsu et al. 2021)



Also employed by SpeechStew (Chan et al. 2021), BigSSL (Zhang et al. 2021)

Multimodal Pretext Tasks

- MIL-NCE (Miech et al. 2020)
  - Find matching narration with video



(Miech et al. 2020)

- CLIP (Radford et al. 2021), ALIGN (Jia et al. 2021)
  - Contrast text and image embeddings from paired data

Language Pretext Tasks: Generative Language Modeling

Pretrained language models: They all rely on unsupervised text and try to predict one sentence from the context.

- GPT: Autoregressive; predict the next token based on the previous tokens.
- BERT: Masked language modeling (MLM) & Next sentence prediction (NSP)
- ALBERT: Sentence order prediction (SOP)
- ELECTRA: Replaced token detection (RTD)

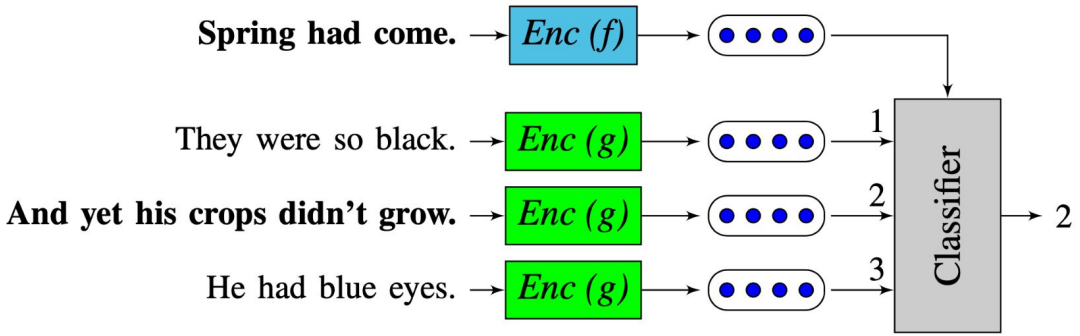
Language Pretext Tasks: Sentence Embedding

Skip-thought vectors (Kiros et al. 2015)

- Predict sentences based on other sentences around.

Quick-thought vectors (Logeswaran & Lee 2018)

- Identify the correct context sentence among other contrastive sentences.



(Quick-thought vectors; Logeswaran & Lee 2018)

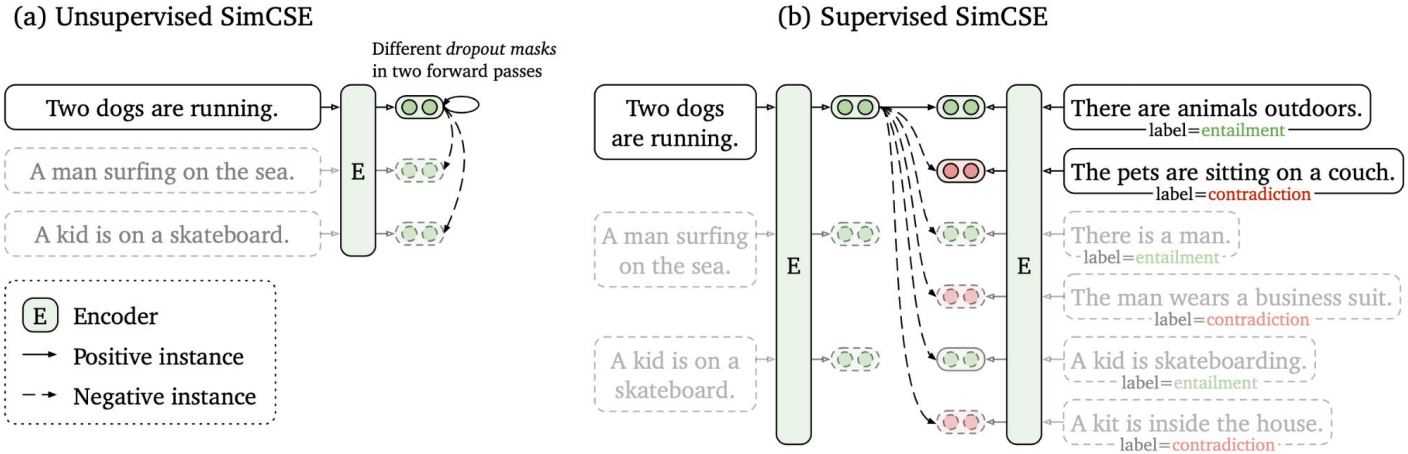
Language Pretext Tasks: Sentence Embedding

IS-BERT (“Info-Sentence BERT”; Zhang et al. 2020)

- mutual information maximization

SimCSE (“Simple Contrastive learning of Sentence Embeddings”; Gao et al. 2021)

- Predict a sentence from itself with only dropout noise.
- One sentence gets two different versions of dropout augmentations.



Language Pretext Tasks: Sentence Embedding

Most of the models for learning sentence embedding relies on supervised NLI (Natural Language Inference) datasets, such as SBERT (Reimers & Gurevych 2019), BERT-flow (Li et al. 2020), Whitening SBERT (Su et al. 2021).

Unsupervised sentence embedding models (e.g. unsupervised SimCSE) still have performance gap with the supervised version (e.g. supervised SimCSE).

Techniques

- Data augmentation
- In-batch negative samples
- Hard negative mining
- Memory bank
- Large batch size

Techniques: Data Augmentation

Data augmentation setup is critical for learning good embedding.

It introduces the non-essential variations into examples without modifying semantic meanings and thus encourages the model to learn the essential part within the representation.

- Image augmentation
- Text augmentation

Techniques: Image Augmentation

- Basic Image Augmentation
  - Random crop
  - color distortion
  - Gaussian blur
  - color jittering
  - random flip/rotation
  - etc.
- Augmentation Strategies
- Image Mixture

Techniques: Image Augmentation

- Basic Image Augmentation
- Augmentation Strategies
  - AutoAugment (Cubuk, et al. 2018): inspired by NAS
  - RandAugment (Cubuk et al. 2019): reduces NAS search space in AutoAugment.
  - PBA (Population based augmentation; Ho et al. 2019): evolutionary algorithm
  - UDA (Unsupervised Data Augmentation; Xie et al. 2019): minimize the KL divergence between the predicted distribution over an unlabelled example and its unlabelled augmented version.
- Image Mixture

Techniques: Image Augmentation

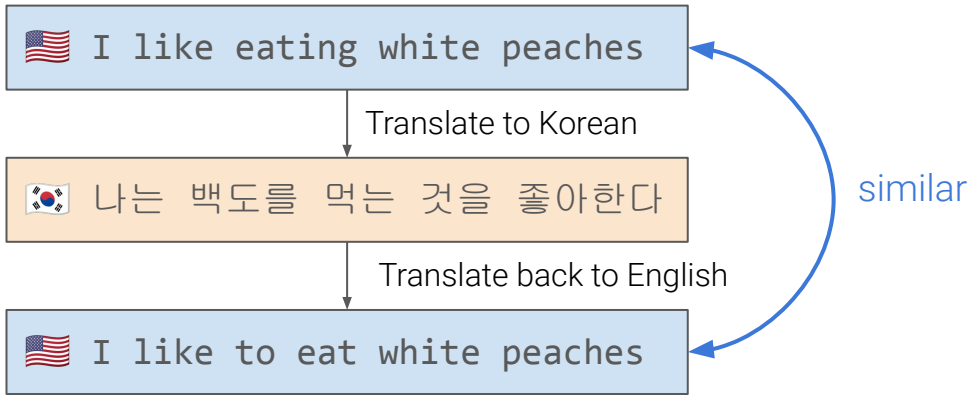
- Basic Image Augmentation
- Augmentation Strategies
- Image Mixture
  - Mixup (Zhang et al 2018): weighted pixel-wise combination of two images.
  - Cutmix (Yun et al 2019): mix in a local region of one image into the other.
  - MoCHi (“Mixing of Contrastive Hard Negatives”; Kalantidis et al 2020): mixture of hard negative samples.

Techniques: Text Augmentation

- Lexical Edits
  - EDA (Easy Data Augmentation; Wei & Zou 2019): synonym replacement, random insertion/swap/deletion.
  - Contextual Augmentation (Kobayashi 2018): word substitution by BERT prediction.
- Back-translation
- Dropout and Cutoff

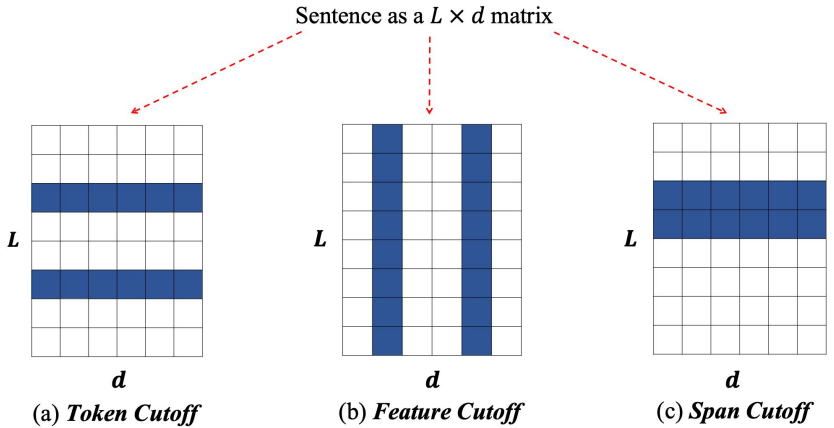
Techniques: Text Augmentation

- Lexical Edits
- **Back-translation** (Sennrich et al. 2015)
  - Back-translation augments one sentence by first translating it to another language and then translating it back to the original language.
  - CERT (Fang et al. 2020) generates augmented sentences via back-translation.
- Dropout and Cutoff



Techniques: Text Augmentation

- Lexical Edits
- Back-translation
- Dropout and Cutoff
  - SimCSE uses dropout (Gao et al. 2021)
  - Cutoff augmentation for text (Shen et al. 2020): tokens, feature columns, spans.



(Shen et al 2020)

Hard Negative Mining

Hard negative samples are different to learn. They should have different labels from the anchor sample, but the embedding features may be very close.

Hard negative mining is important for contrastive learning.

Challenging negative samples encourages the model to learn better representations that can distinguish hard negatives from true positives.

Hard Negative Mining

Explicit hard negative mining

- Extract task-specific hard negative samples from labelled datasets.
  - e.g. "`contradiction`" sentence pairs from NLI datasets. (Most sentence embedding papers)
- Keyword based retrieval
  - e.g. BM25 search results (Karpukhin et al. 2020)
- Upweight the negative sample probability to be proportional to its similarity to the anchor sample (Robinson et al. 2021)
- MoCHi (Kalantidis et al. 2020): mine hard negative by sorting them according to similarity to the query in descending order.

Hard Negative Mining

Implicit hard negative mining

- In-batch negative samples
- Memory bank (Wu et al. 2018, He et al. 2019) → Increase batch size
- Large batch size via various training parallelism

Theories

Why does contrastive learning work?

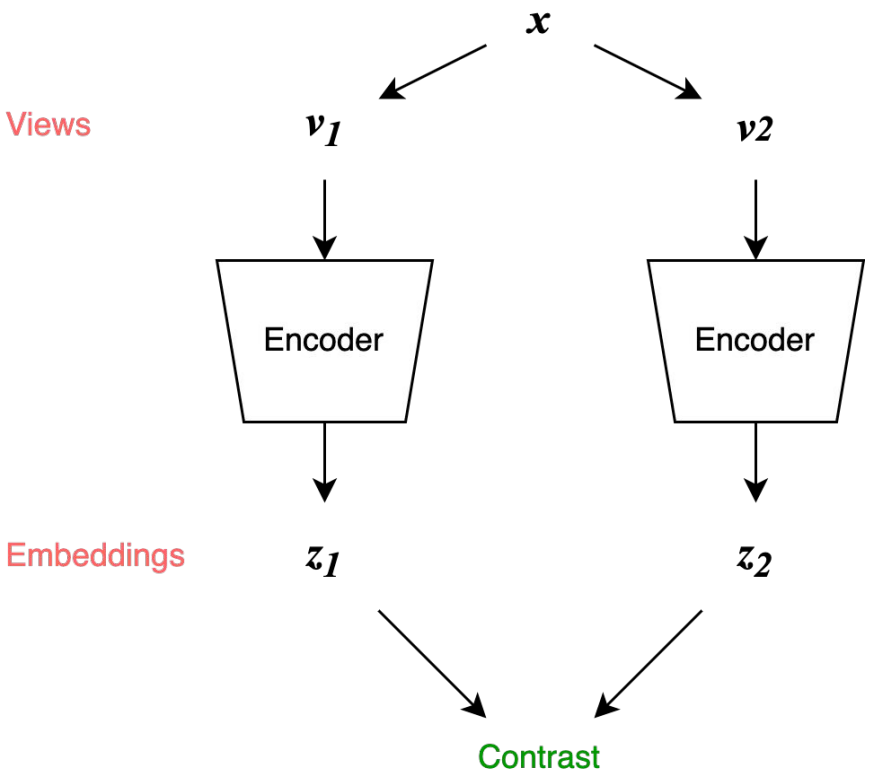
Contrastive learning captures shared information between views

InfoNCE (van den Oord et al. 2018) is a lower bound to MI between views:

$$I(\mathbf{v}_1; \mathbf{v}_2) \geq I(\mathbf{z}_1; \mathbf{z}_2) \geq \log(\mathrm{K}) - \mathcal{L}_{\mathrm{InfoNCE}}$$

Minimizing InfoNCE leads to maximizing the MI between view 1 and view 2.

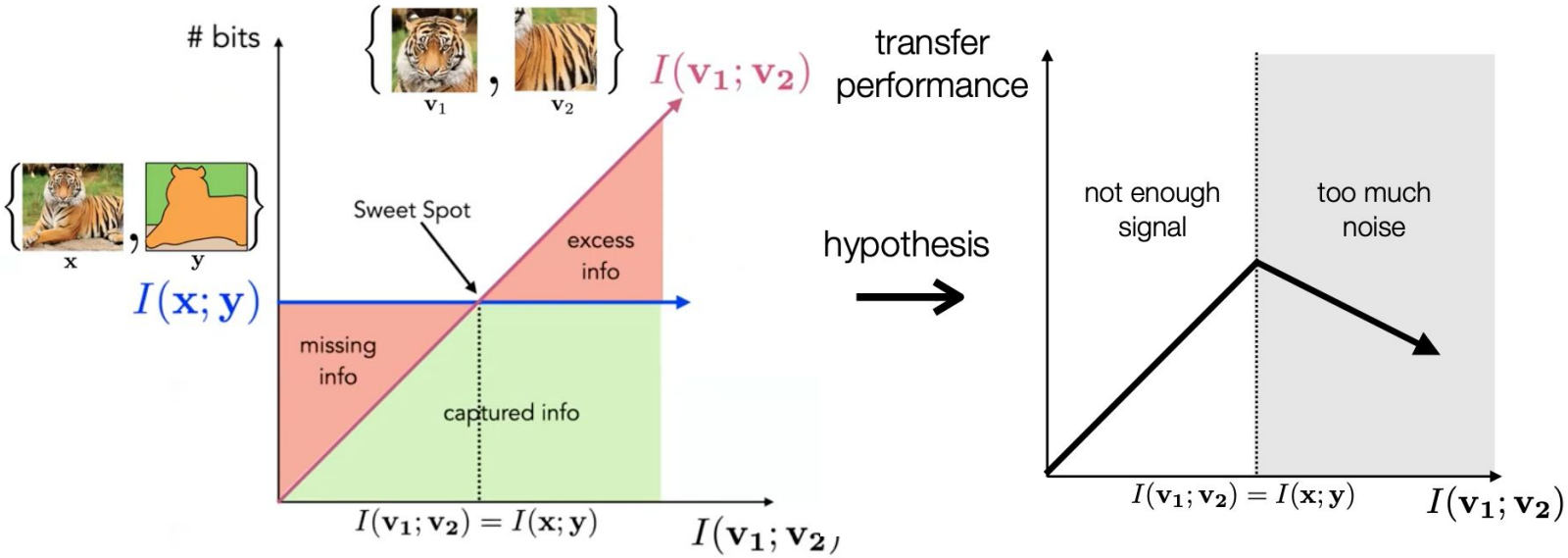
Q: How can we design good views?



The InfoMin Principle

Optimal views are at the sweet spot where it only encodes useful information for transfer

- Minimal sufficient encoder depends on downstream tasks (Tian et al. 2020)
- Composite loss for finding the sweet spot (Tsai et al. 2020)

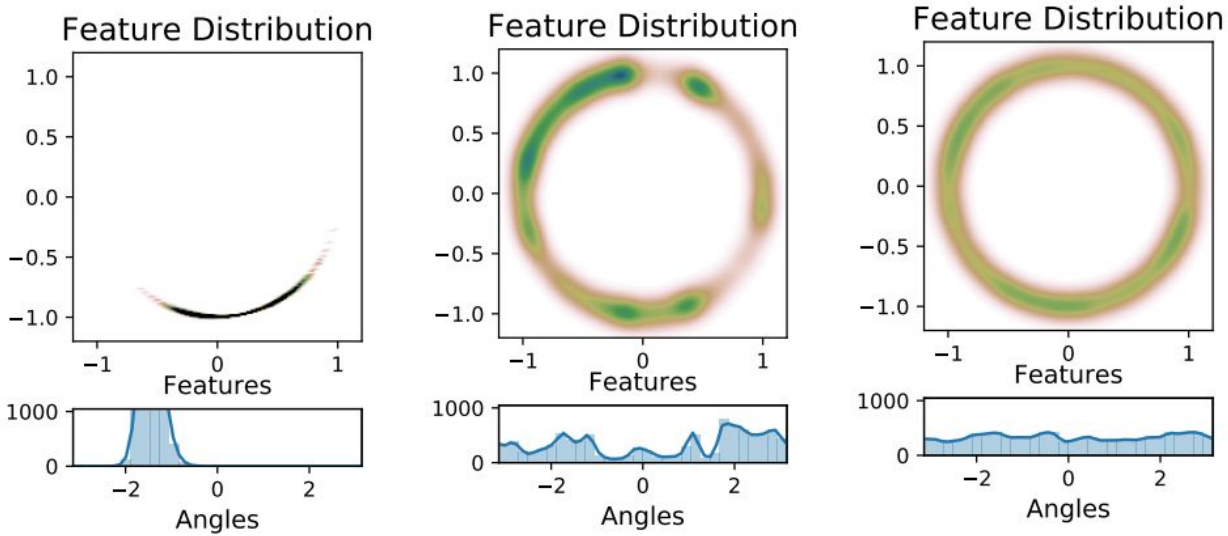


(Tian et al. 2020)

Alignment and Uniformity on the Hypersphere

Contrastively learned features are more **uniform** and **aligned**.

- **Uniform**: features should be distributed uniformly on the hypersphere S^d
- **Aligned**: features from two views of the same input should be the same



Random init. Supervised Contrastive

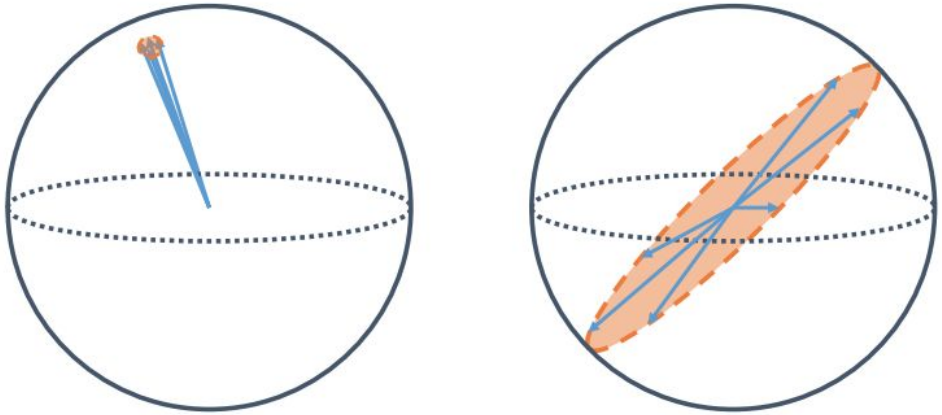
Dimensional Collapse

Contrastive methods sometimes suffer from dimensional collapse (Hua et al. 2021)

- Features span lower-dimensional subspace instead

Two causes demonstrated by Jing et al. (2021)

- Strong augmentation & implicit regularization



(Jing et al. 2021)

Complete Collapse

Dimensional Collapse

Provable Guarantees for Contrastive Learning

Sampling complexity decreases when:

- Adopting contrastive learning objectives (Arora et al. 2019)
- Predicting the known distribution in the data (Lee et al. 2020)

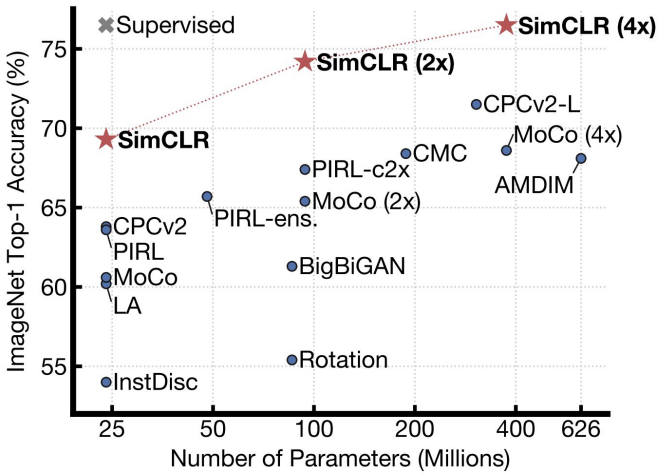
Linear classifier on learned representation is nearly optimal (Tosh et al. 2021)

Spectral Contrastive Learning (HaoChen et al. 2021)

Future directions

Future Directions

- Large batch size → improved transfer performance.
- High-quality large data corpus → Better performance.
  - Learning from synthetic or Web data.
  - Measuring dataset quality and filtering / active learning
- Efficient negative sample selection.
- Combine multiple pretext tasks.
  - How to combine
  - Best strategies



Future Directions

- Data augmentation tricks have critical impacts but are still quite ad-hoc
  - Modality-dependent
  - Theoretical foundations
- Improving training efficiency
  - Self-supervised learning methods are pushing the deep learning arms race
  - Direct impacts on the economical and environmental costs
- Social biases in the embedding space.
  - Early work in debiasing word embedding.
  - Biases in Dataset

Thank You

Visit openai.com for more information.

FOLLOW @OPENAI ON TWITTER