Georgia Tech College of Computing

CS4641 Spring 2025

Representation Learning

Bo Dai
School of CSE, Georgia Tech
bodai@cc.gatech.edu

Supervised Learning vs. Unsupervised Learning

Training Data

Supervised $\{x^i, y^i\}_{i=1}^m$

Unsupervised $\{x_i\}_{i=1}^m$

Learning Algorithm

Target Function

$$f : X \to Y$$

Density Estimation $p(x)$

Clustering

$$f : X \to Y$$

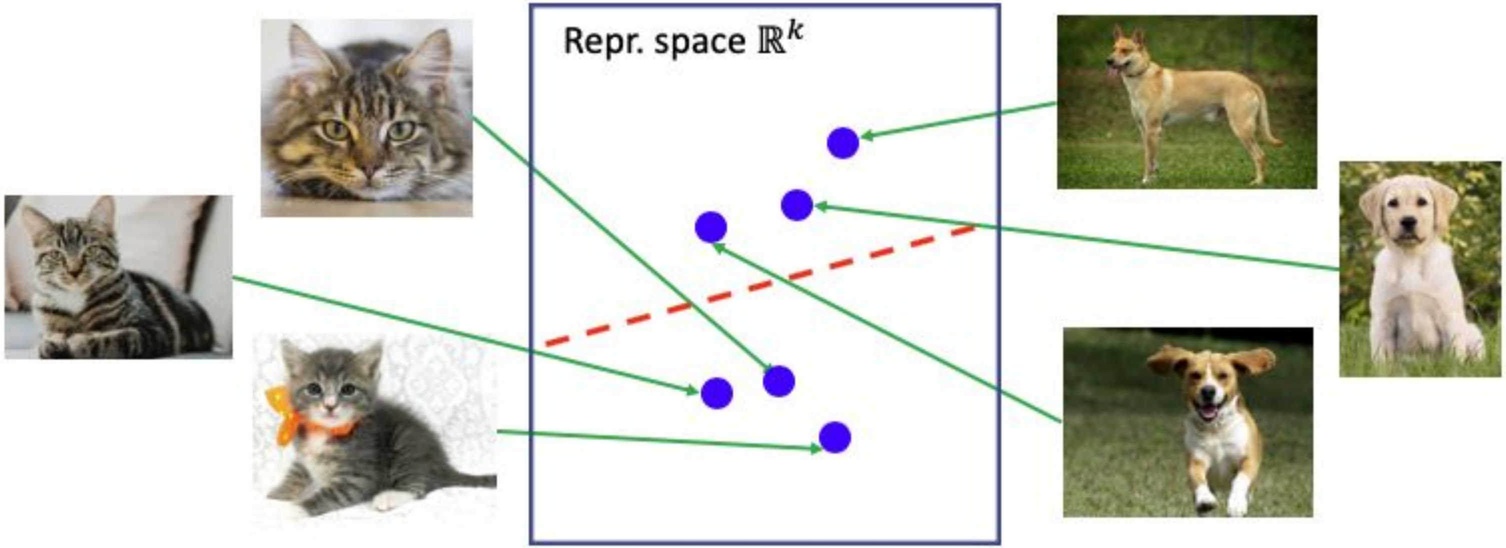
$$Y \in \{0, 1, \ldots, k\}$$

Dimension Reduction

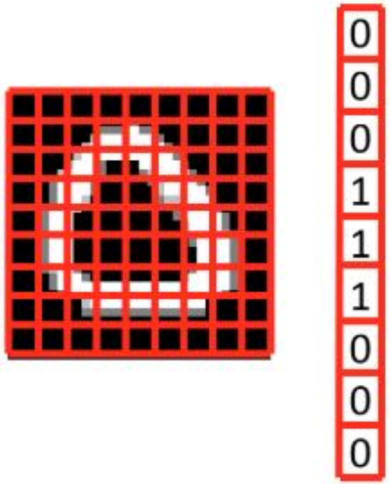
$$f : X \to Y$$

$$X \in \mathbb{R}^p, \quad Y \in \mathbb{R}^d$$

Dimension Reduction/Representation Learning



Dimension Reduction/Representation Learning



vector in $\mathbb{R}^p$

Original data point

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{pmatrix}$$

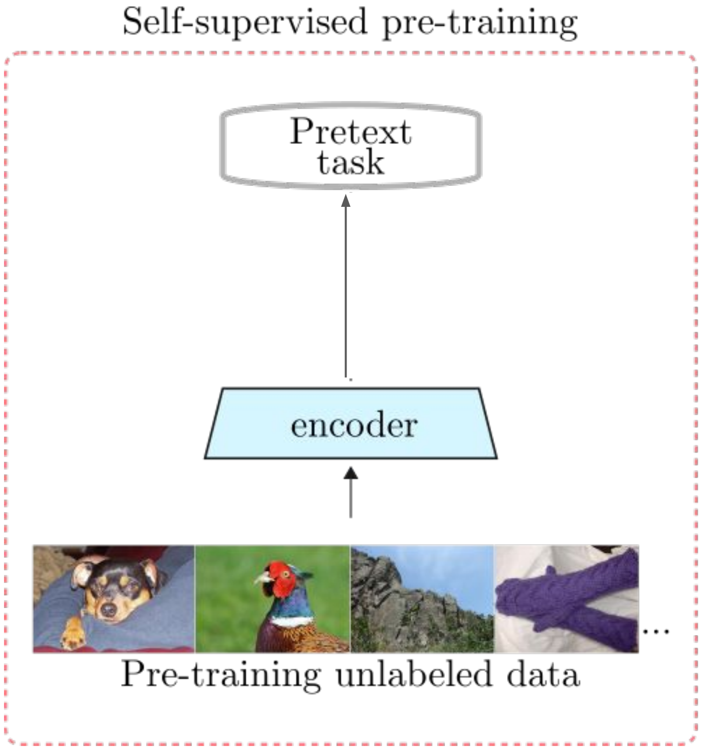
$$f(\mathbf{x}) : \mathbb{R}^p \to \mathbb{R}^d$$

Reduced representation

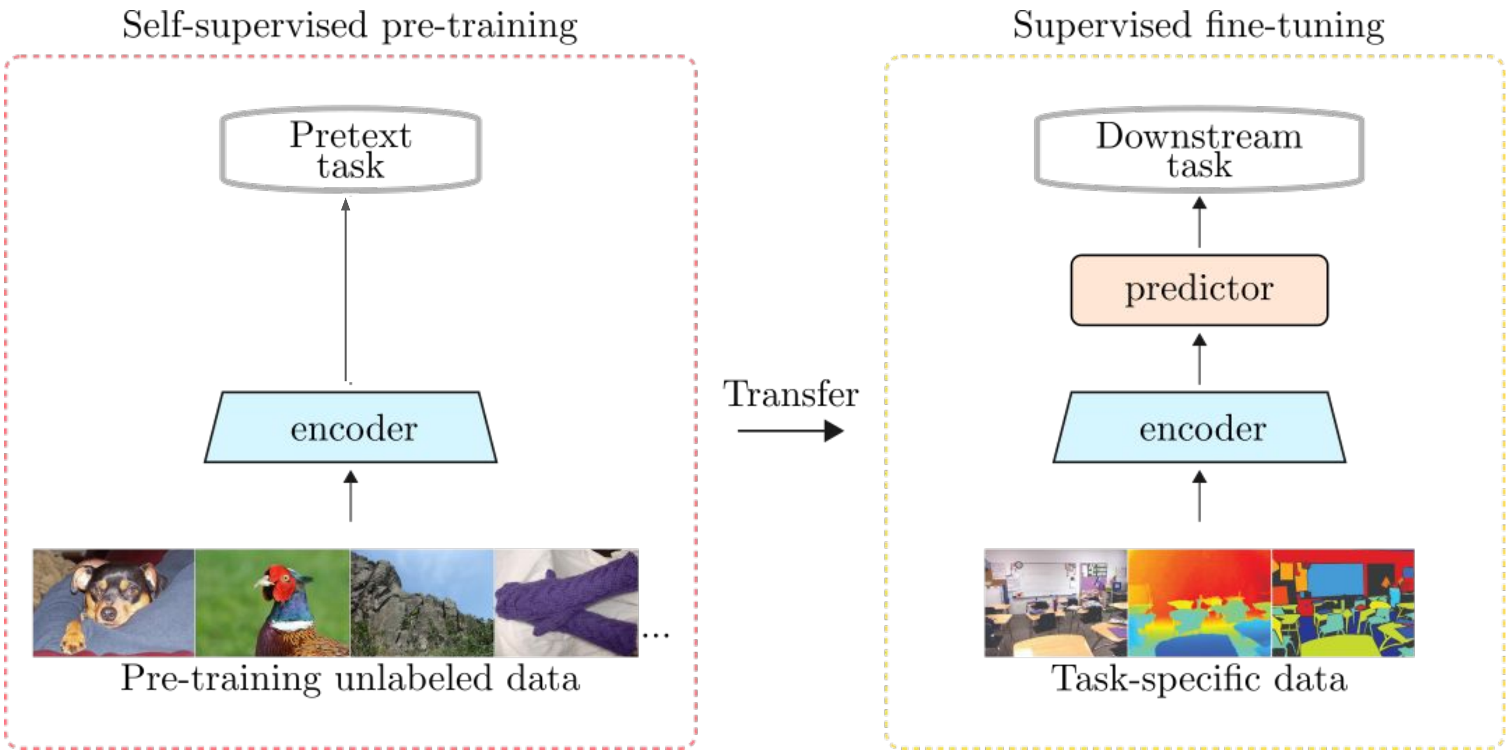
$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_d \end{pmatrix}$$

$$d \ll p$$

Usage of Representation in ML Tasks



Usage of Representation in ML Tasks



Probabilistic Principal Component Analysis as LVM

Training Data

$\{x_i\}_{i=1}^m$

Learning Algorithm

Target Function:

$$f : X \to Y$$

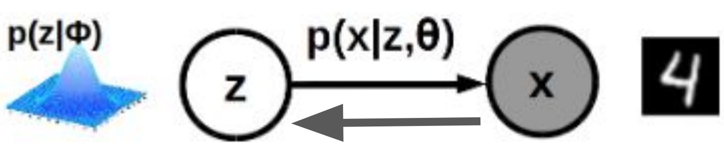
$$X \in \mathbb{R}^p, \quad Y \in \mathbb{R}^d$$

Density Estimation Pipeline

1. Build probabilistic models
   Gaussian Latent Variable Model
2. Derive loss function (by MLE or MAP….)
   MLE
3. Select optimizer
   Necessary Condition

Gaussian LVM for Dimension Reduction

Generation as Pretext Tasks



$$q(z|x)$$

$$p(z) = \mathcal{N}(0, \sigma I)$$

$$p(x|z) = \mathcal{N}(Wz + \mu, \sigma^2 I)$$

$$q(z|x) = \frac{p(x|z)p(z)}{p(x)}$$
$$= \mathcal{N}(MW^\top (x - \mu), \sigma^2 M)$$
$$M = (W^\top W + \sigma^2 I)^{-1}$$

$$p(x) = \int p(x|z)p(z)dz$$

$$p(x) = \mathcal{N}(\mu, WW^\top + \sigma^2 I)$$

- The posterior mean is given by (see the tutorial)

$$E[z|x] = (W^T W + \sigma^2 I)^{-1} W^T (x - \mu)$$

- Posterior variance:

$$\mathrm{Cov}[z|x] = \sigma^2 (W^T W + \sigma^2 I)^{-1}$$

[Chap 2.3 in Pattern Recognition and Machine Learning]

$$\sum_{n=1}^{N} C^{-1}(x_n - \mu) = 0 \quad \Rightarrow \quad \mu = \frac{1}{N}\sum_{n=1}^{N} x_n$$

- The optimal parameters for the maximal log-likelihood are

$$\mu = \frac{1}{N}\sum_{n=1}^{N} x_n$$

$$\sigma^2 = \frac{1}{D-M}\sum_{j=M+1}^{D} \lambda_j$$

$$W = U_M(\Lambda_M - \sigma^2 I)^{1/2}$$

Denote

$$S = S_{true} + S_{noise}$$

$$U\Lambda U^T = U_M \Lambda_M U_M^T + U_n \Lambda_n U_n^T$$

Covariance of Gaussian was

$$C = WW^T + \sigma^2 I$$

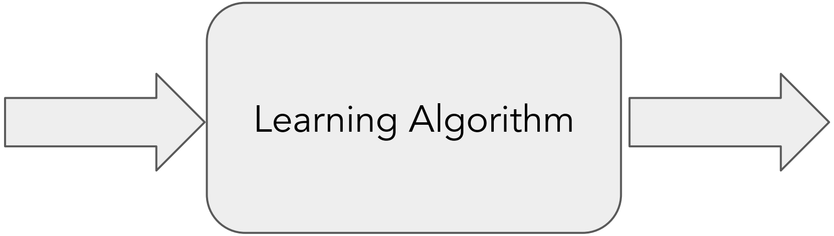
C should match S_{true}

$$U_M \Lambda_M U_M^T = WW^T + \sigma^2 I$$

Latent Variable Model for Representation Learning

Training Data

$\{x_i\}_{i=1}^m$



Target Function:

$$f : X \to Y$$

$$X \in \mathbb{R}^p, \quad Y \in \mathbb{R}^d$$

Density Estimation Pipeline

1. Build probabilistic models
   Deep Latent Variable Model
2. Derive loss function (by MLE or MAP….)
   ELBO
3. Select optimizer
   Stochastic Gradient Descent

Revisit Latent Variable Models

Generation as Pretext Tasks

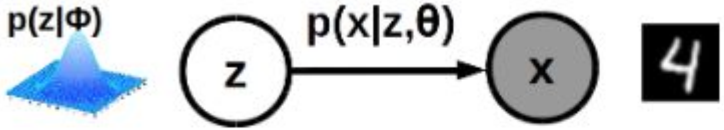
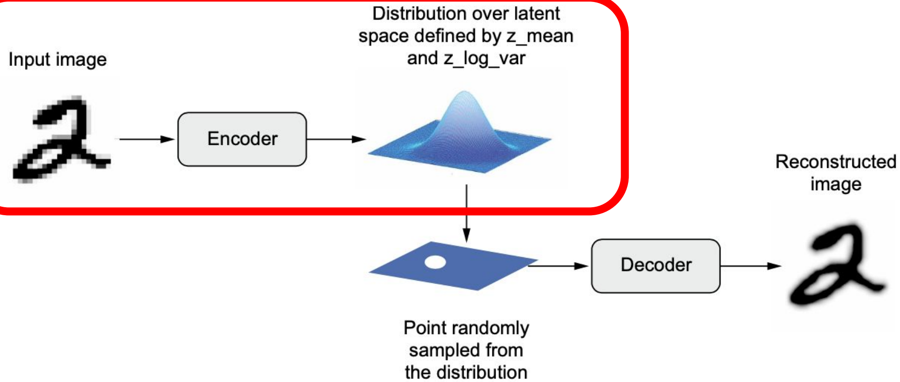


Figure 5: 1024 × 1024 images generated using the CELEBA-HQ dataset. See Appendix F for a larger set of results, and the accompanying video for latent space interpolations.

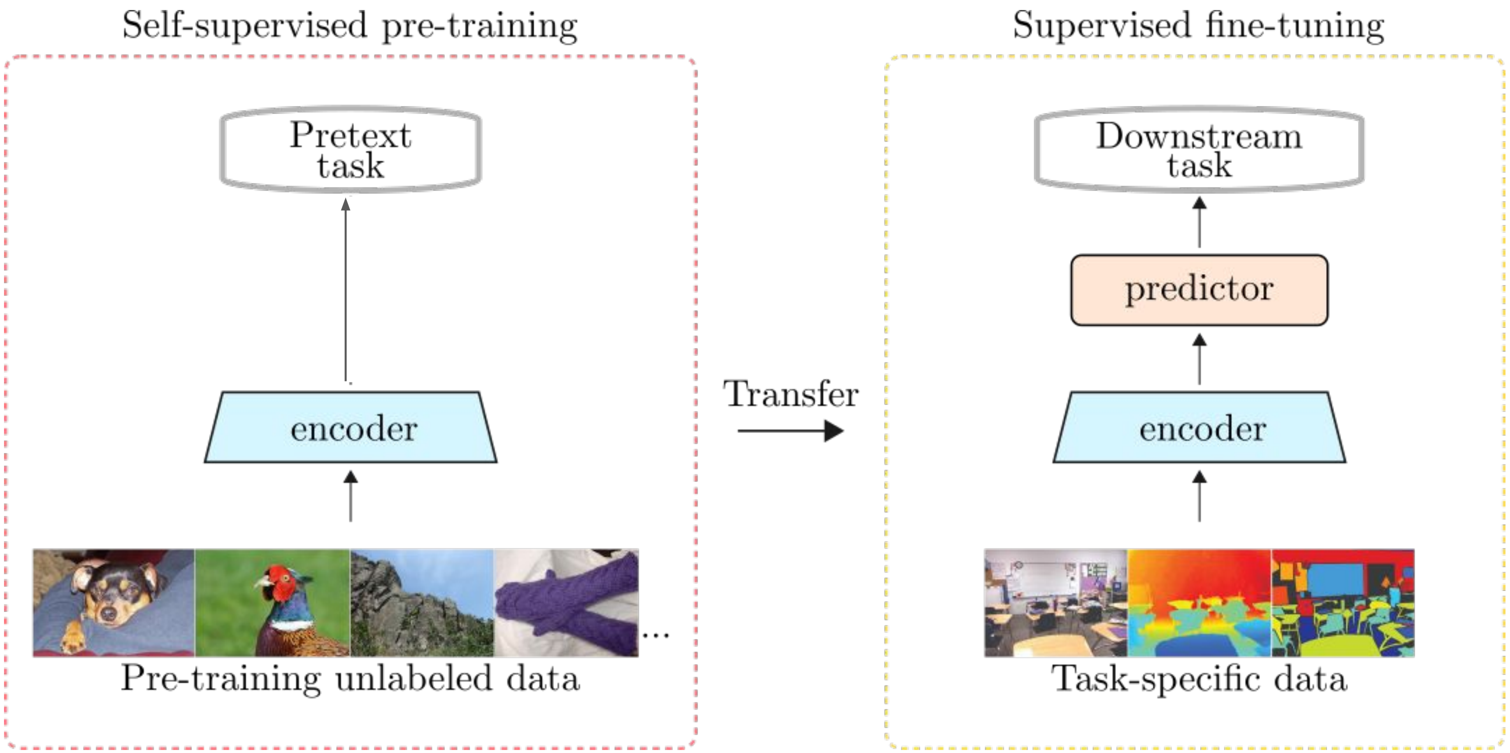
$$p(x) = \int p(x|z)p(z)dz$$

Generalized LVM for Representation Learning

$$\boxed{\max_{W_z^1, W_z^2}} \max_{\sigma, W_\sigma, W_\mu} \sum_{i=1}^{m} \int q(z^i|x^i)\Big(\log p(x^i|z^i)p(z^i)\Big)dz^i \underbrace{- \int q(z^i|x^i)\log q(z^i|x^i)dz_i}_{H(q(z|x))}$$



Another Pretext Tasks?



Contrastive Representation Learning

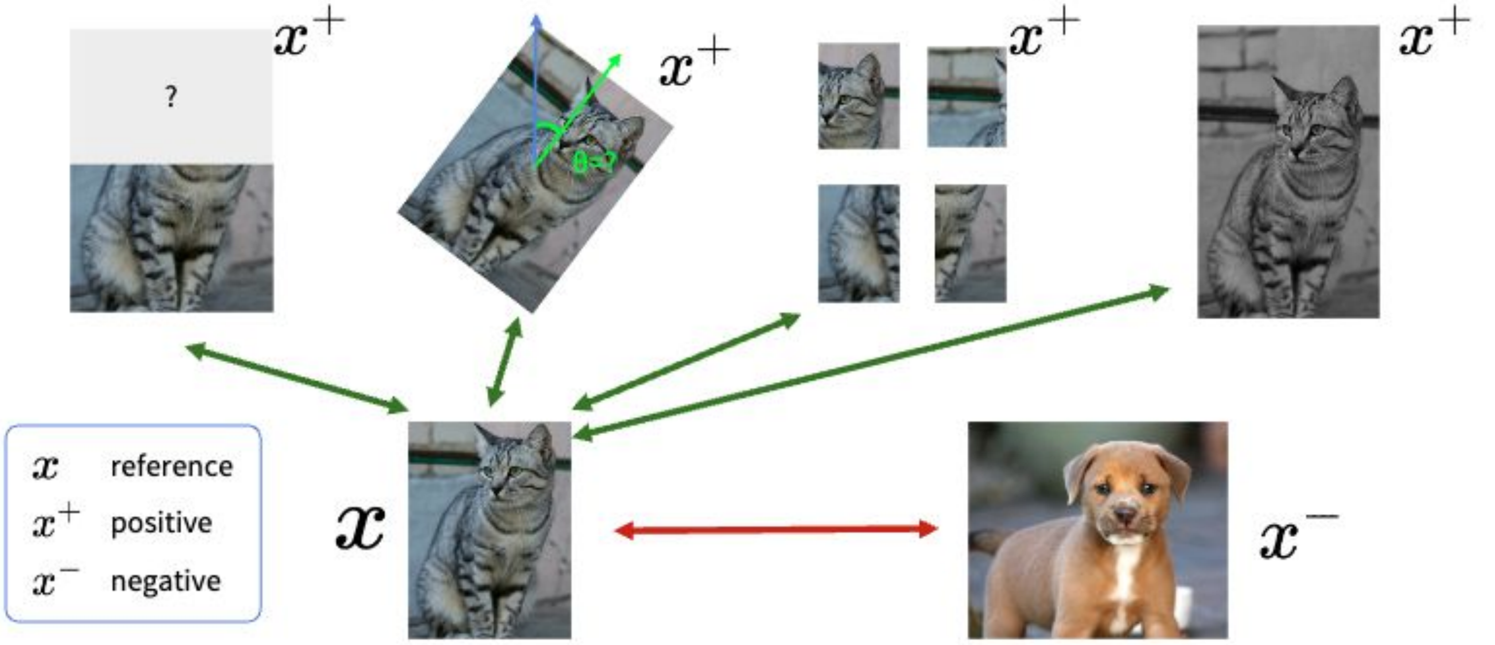
- Basic Idea - Design Pretext Tasks
    - Convert the unsupervised learning to supervised learning

Contrastive Representation Learning

- Basic Idea - Design Pretext Tasks
  - Convert the unsupervised learning to supervised learning

1. Synthesis labels
2. Apply the supervised methods to the synthesis labels
3. Extract the representation

Synthesis Labels



Supervised Tasks

Training Data

$$\{x_i, x_i^j, y_{ij}\}_{i=1,j=1}^{n,k}$$

$$y_{ij} \in \{0, 1\}$$

Learning Algorithm

Target Function:

$$f : X \times X \to Y$$

Supervised Tasks: Binary Classification

Training Data

$$\{\underbrace{x_i, x_i^j}_{x'_{ij}}, y_{ij}\}_{i=1,j=1}^{n,k} \quad y_{ij} \in \{0, 1\}$$

Learning Algorithm

Target Function:

$$f : \underbrace{X \times X}_{X'} \to Y$$

Supervised Tasks: Binary Classification

Training Data

$$\{\underbrace{x_i, x_i^j}_{x'_{ij}}, y_{ij}\}_{i=1,j=1}^{n,k} \quad y_{ij} \in \{0, 1\}$$

Learning Algorithm

Target Function:

$$f : \underbrace{X \times X}_{X'} \to Y$$

Logistic Regression Pipeline

1. Build probabilistic models: Bernoulli Distribution
2. Derive loss function: MLE and MAP
3. Select optimizer: (Stochastic) Gradient Descent

Probabilistic Model in Classification: Bernoulli Likelihood

$$p(y) = p^y(1-p)^{(1-y)} \qquad p \in [0,1]$$

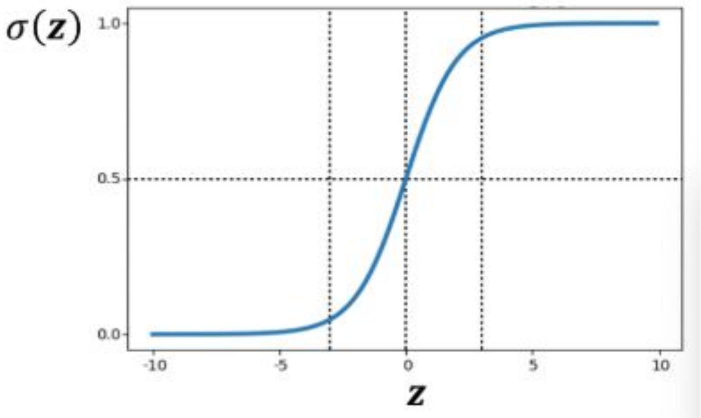
$$p(y|x') = p(y=1|x')^y\{1 - p(y=1|x')\}^{1-y}$$

Probabilistic Model in Classification: Bernoulli Likelihood

$$\sigma(\mathbf{z}) = \frac{1}{1+e^{-\mathbf{z}}} = \frac{e^{\mathbf{z}}}{1+e^{\mathbf{z}}}$$

$$\phi : X \to S$$

$$X \in \mathbb{R}^d, \quad S \in \mathbb{R}^p$$



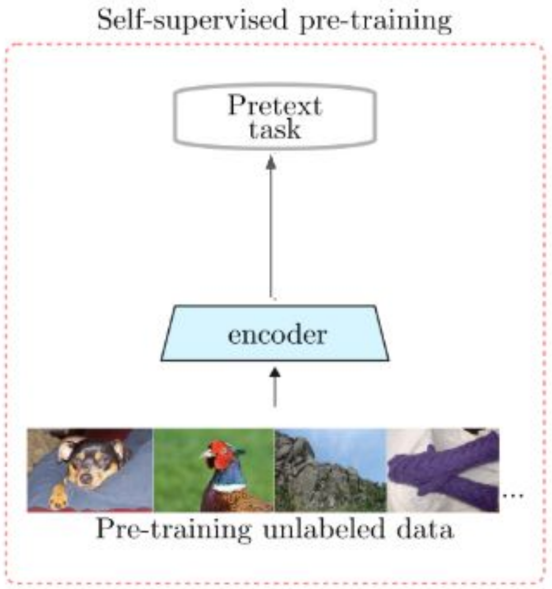
$$p(y=1|x') = \sigma(\phi(x_1)^\top \phi(x_2)) = \frac{1}{1+\exp(-\phi(x_1)^\top \phi(x_2))}$$

$$p(y=0|x') = 1 - \frac{1}{1+\exp(-\phi(x_1)^\top \phi(x_2))} = \frac{\exp(-\phi(x_1)^\top \phi(x_2))}{1+\exp(-\phi(x_1)^\top \phi(x_2))}$$

Where is Representation?

$$p(y = 1|x') = \frac{1}{1 + \exp(-\phi(x_1)^\top \phi(x_2))}$$

$$\phi : X \to S$$
$$X \in \mathbb{R}^d, \quad S \in \mathbb{R}^p$$



Supervised Tasks: Binary Classification

Training Data

$$\{\underbrace{x_i, x_i^j}_{x'_{ij}}, y_{ij}\}_{i=1,j=1}^{n,k} \quad y_{ij} \in \{0, 1\}$$

Learning Algorithm

Target Function:

$$f : \underbrace{X \times X}_{X'} \to Y$$

Logistic Regression Pipeline

1. Build probabilistic models: Bernoulli Distribution
2. Derive loss function: MLE and MAP
3. Select optimizer: (Stochastic) Gradient Descent

MLE

- Logistic regression model

$$p(y=1|x') = \frac{1}{1+\exp(-\phi(x_1)^\top \phi(x_2))}$$

- Plug in

$$\begin{aligned}
\ell(\theta) &:= \log \prod_{i=1}^{n} p(y^i \mid \phi(x_2^i), \phi(x_1^i)) \\
&= \sum_{i=1}^{n} \log\left(\frac{\exp(-\phi(x_1^i)^\top \phi(x_2^i))}{1+\exp(-\phi(x_1^i)^\top \phi(x_2^i))}\right) \cdot \underbrace{I(y^i=0)}_{1-y^i} + \log\left(\frac{1}{1+\exp(-\phi(x_1^i)^\top \phi(x_2^i))}\right) \cdot \underbrace{I(y^i=1)}_{y^i} \\
&= \sum_{i=1}^{n} \left((y^i - 1) \cdot \phi(x_1^i)^\top \phi(x_2^i) - \log\left(1+\exp\left(-\phi(x_1^i)^\top \phi(x_2^i)\right)\right)\right)
\end{aligned}$$

Supervised Tasks: Binary Classification

Training Data

$$\{\underbrace{x_i, x_i^j}_{x'_{ij}}, y_{ij}\}_{i=1,j=1}^{n,k} \quad y_{ij} \in \{0, 1\}$$

Learning Algorithm

Target Function:

$$f : \underbrace{X \times X}_{X'} \to Y$$

Logistic Regression Pipeline

1. Build probabilistic models: Bernoulli Distribution
2. Derive loss function: MLE and MAP
3. Select optimizer: (Stochastic) Gradient Descent

Gradient Calculation of MLE

$$\max_{\phi} \log L(\phi) = \sum_{i} (y^i - 1)\, \phi(x_1^i)^\top \phi(x_2^i) - \log\left(1 + \exp\left(-\phi(x_1^i)^\top \phi(x_2^i)\right)\right)$$

(Stochastic) Gradient Descent

- Initialize parameter ϕ

- Do

$$\phi^{t+1} \leftarrow \phi^{t} + \eta \nabla \ell(\phi)$$

Supervised Task: Binary Classification

Training Data

$$\{\underbrace{x_i, x_i^j}_{x'_{ij}}, y_{ij}\}_{i=1,j=1}^{n,k} \quad y_{ij} \in \{0, 1\}$$

Learning Algorithm

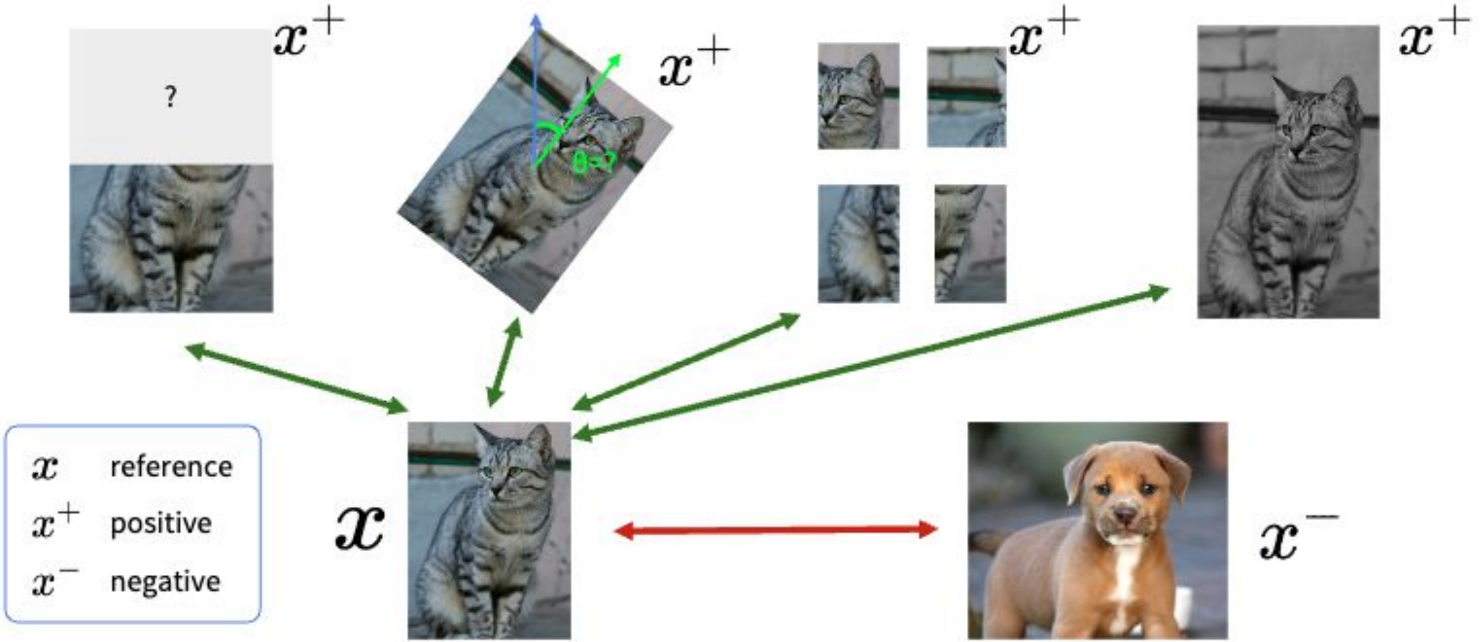
Target Function:

$$f : \underbrace{X \times X}_{X'} \to Y$$

Logistic Regression Pipeline

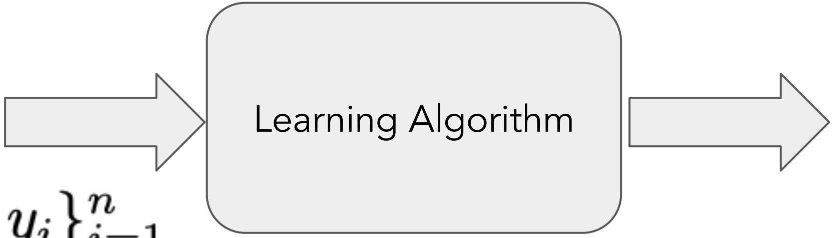
1. Build probabilistic models: Bernoulli Distribution
2. Derive loss function: MLE and MAP
3. Select optimizer: (Stochastic) Gradient Descent

Synthesis Labels



Supervised Tasks: Multiclass Classification

Training Data



Target Function:

$$\{\underbrace{x_i, x_i^+, x_i^1, \ldots, x_i^k}_{x_i'}, y_i\}_{i=1}^n$$

$$f : \underbrace{X^{k+2}}_{X'} \rightarrow Y$$

$$y_i = [0, 1, 0, \ldots, 0]$$

Multiclass Logistic Regression Pipeline

1. Build probabilistic models: Categorical Distribution
2. Derive loss function: MLE and MAP
3. Select optimizer: (Stochastic) Gradient Descent

Probabilistic Model in Multiclass Classification:
Categorical Likelihood

$$p(y = i) = p_i, \quad \sum_{i=1}^{k} p_i = 1, \quad p_i \geq 0$$

$$p(y) = \prod_{i=1}^{k} p_i^{y_i}$$

$$p = (p_1, p_2, \ldots, p_k)$$

$$y = (y_1, y_2, \ldots, y_k), \quad y_i \in 0, 1, \quad \sum_{i=1}^{k} y_i = 1$$

1-of-k code

Softmax Parametrization

$$p(y^2 = 1|x') = \frac{\exp(\phi(x)^\top \phi(x^+))}{\exp(\phi(x)^\top \phi(x^+)) + \sum_{j=1}^{k} \exp(\phi(x)^\top \phi(x^j))}$$

Supervised Tasks: Multiclass Classification

Training Data

Learning Algorithm

Target Function:

$$\underbrace{\{x_i, x_i^+, x_i^1, \dots, x_i^k}_{x_i'}, y_i\}_{i=1}^n$$

$$f : \underbrace{X^{k+2}}_{X'} \to Y$$

$$y_i = [0, 1, 0, \dots, 0]$$

Multiclass Logistic Regression Pipeline

1. Build probabilistic models: Categorical Distribution
2. Derive loss function: MLE and MAP
3. Select optimizer: (Stochastic) Gradient Descent

MLE

$$\max_{\phi} \log L(\phi) = \log \prod_{i=1}^{n} \prod_{j=0}^{k} p(y_i^j | x_i')$$

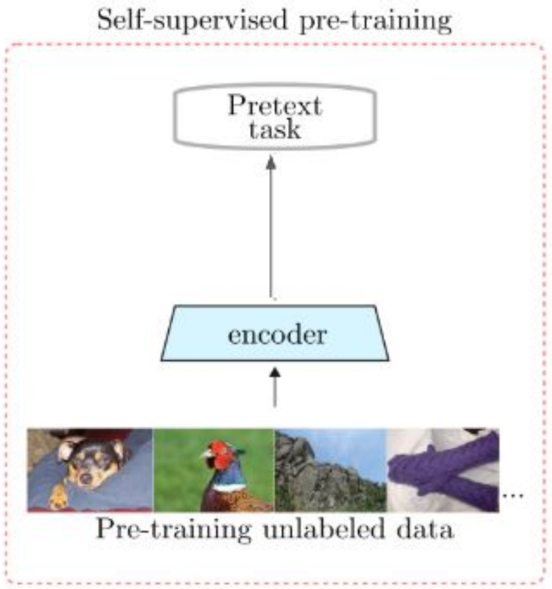
$$= \sum_{i=1}^{n} \log \frac{\exp(\phi(x_i)^\top \phi(x_i^+))}{\exp(\phi(x_i)^\top \phi(x_i^+)) + \sum_{j=1}^{k} \exp(\phi(x_i)^\top \phi(x_i^j))}$$

Where is Representation?

$$p(y^2 = 1|x') = \frac{\exp(\phi(x)^\top \phi(x^+))}{\exp(\phi(x)^\top \phi(x^+)) + \sum_{j=1}^{k} \exp(\phi(x)^\top \phi(x^j))}$$

$$\phi : X \to S$$

$$X \in \mathbb{R}^d, \quad S \in \mathbb{R}^p$$



Gradient of MLE

$$\sum_{i=1}^{n} \log \frac{\exp(\phi(x_i)^\top \phi(x_i^+))}{\exp(\phi(x_i)^\top \phi(x_i^+)) + \sum_{j=1}^{k} \exp(\phi(x_i)^\top \phi(x_i^j))}$$

SimCLR (ICML 2020): Multiclass Classification

Training Data → Learning Algorithm → Target Function:

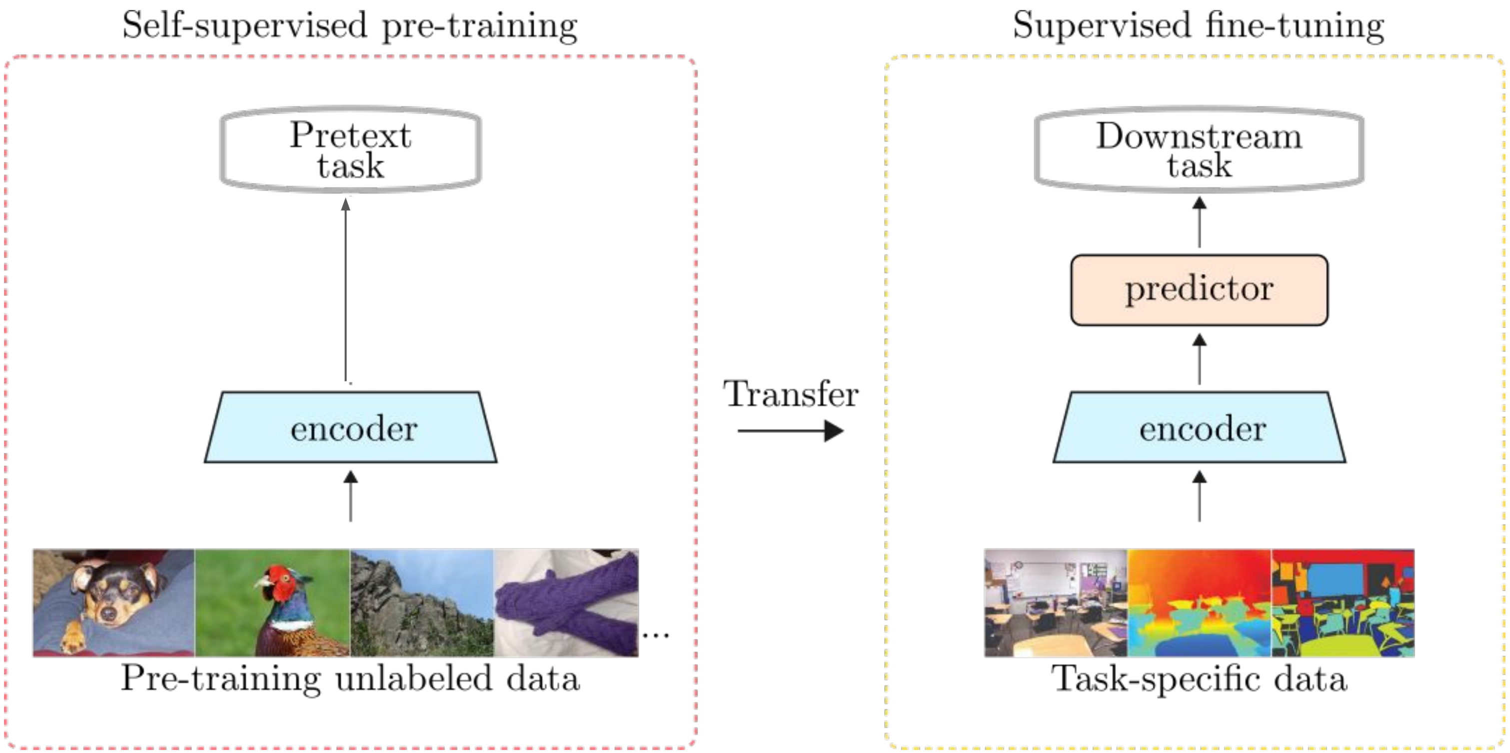
$$\underbrace{\{x_i, x_i^+, x_i^1, \ldots, x_i^k}_{x_i'}, y_i\}_{i=1}^n \qquad y_i = [0, 1, 0, \ldots, 0]$$

$$f : \underbrace{X^{k+2}}_{X'} \to Y$$

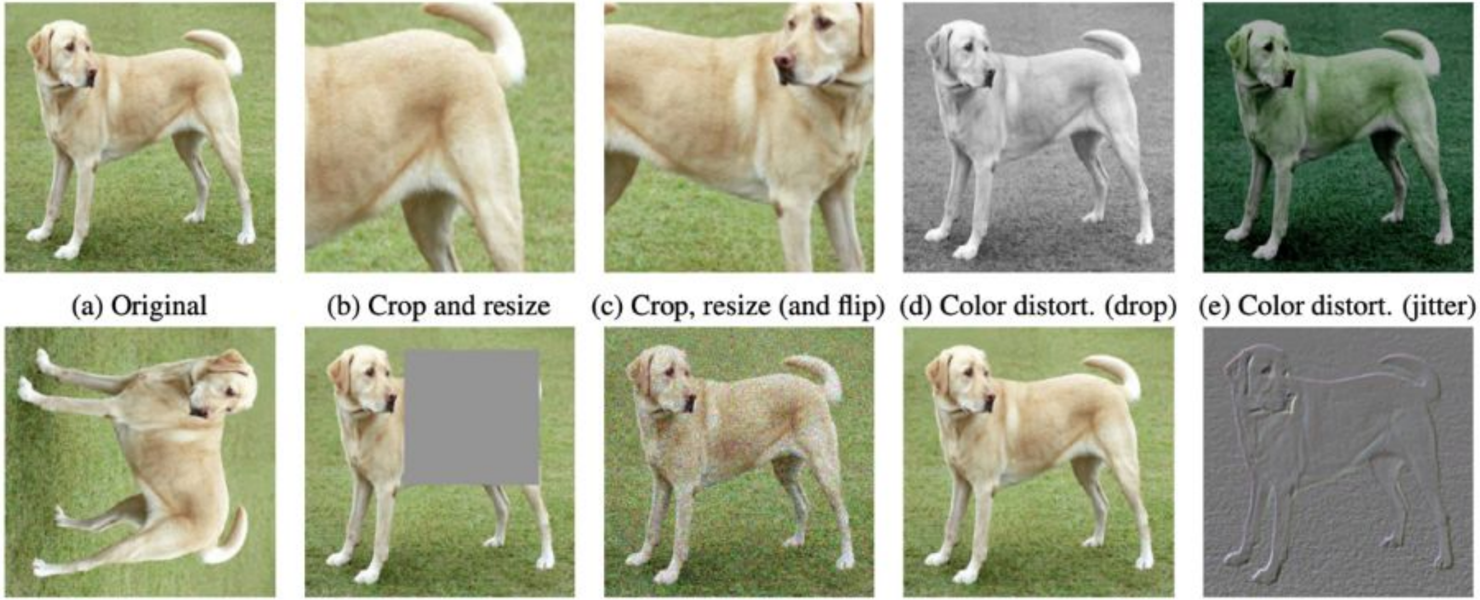
Multiclass Logistic Regression Pipeline

1. Build probabilistic models: Categorical Distribution
2. Derive loss function: MLE and MAP
3. Select optimizer: (Stochastic) Gradient Descent

Usage of Representation in ML Tasks



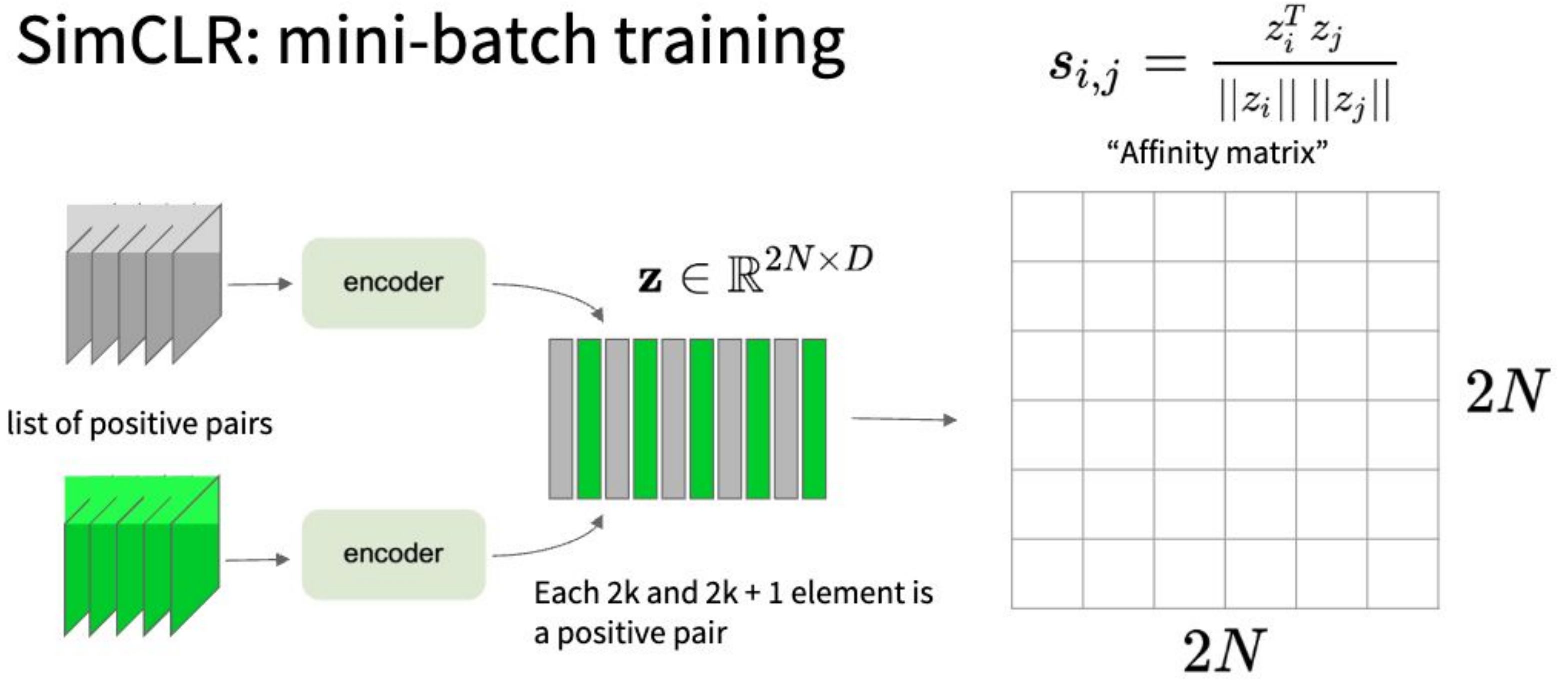
Some Details: Positive Samples



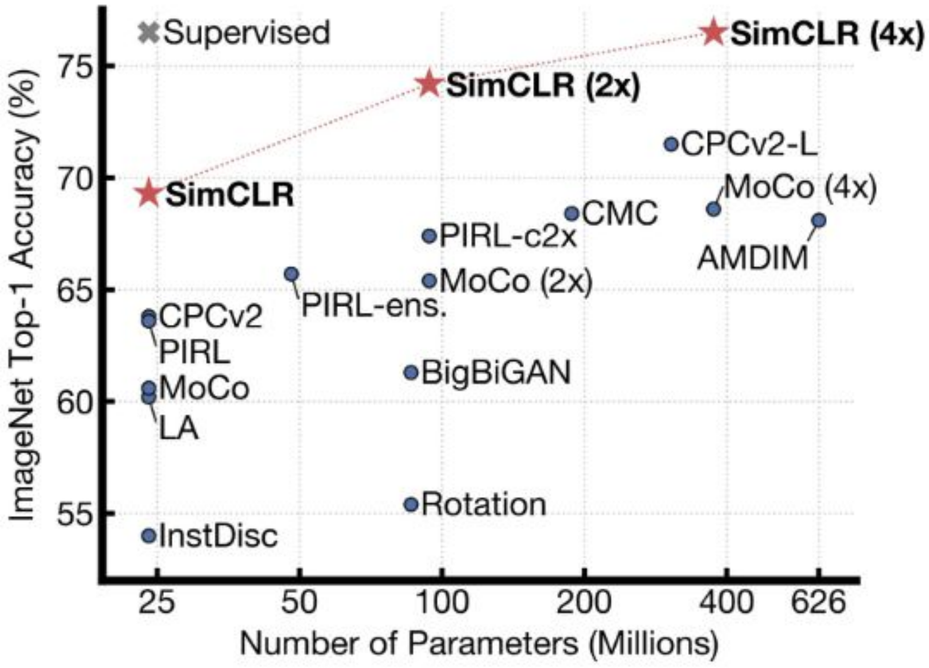
(a) Original (b) Crop and resize (c) Crop, resize (and flip) (d) Color distort. (drop) (e) Color distort. (jitter)

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

Some Details: Negative Samples



Empirical Performances



Train feature encoder on ImageNet
(entire training set) using SimCLR.

Freeze feature encoder, train a linear classifier on top with labeled
data.

Extension

Training Data → Learning Algorithm → Target Function:

$$\underbrace{\{x_i, x_i^+, x_i^1, \ldots, x_i^k}_{x_i'}, y_i\}_{i=1}^n$$

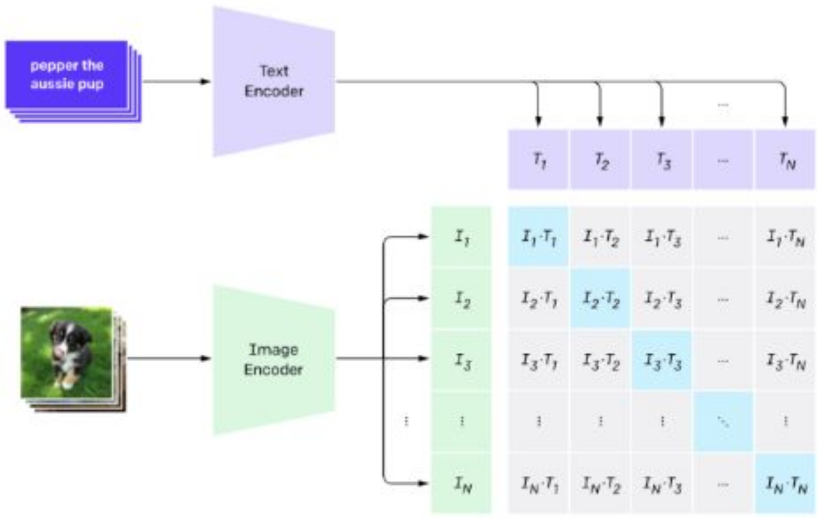
$$y_i = [0, 1, 0, \ldots, 0]$$

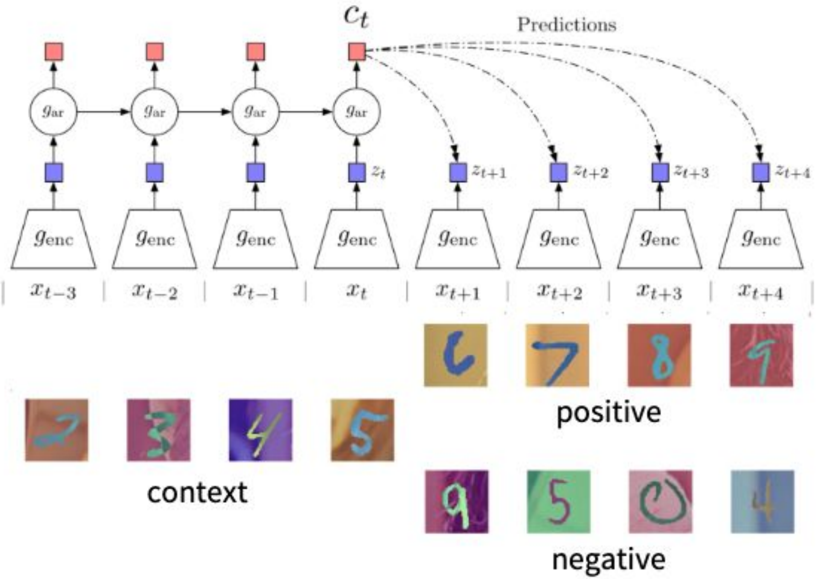
$$f : \underbrace{X^{k+2}}_{X'} \rightarrow Y$$

$$p(y^2 = 1|x') = \frac{\exp(\phi(x)^\top \phi(x^+))}{\exp(\phi(x)^\top \phi(x^+)) + \sum_{j=1}^k \exp(\phi(x)^\top \phi(x^j))}$$

Extension

- Extension: multi-modality (CLIP), sequences (CPC)





Summary

- Representation Learning Pipeline

  Convert the unsupervised learning to supervised learning

  - Synthesis labels
  - Apply the supervised methods to the synthesis labels
  - Extract the representation

- Extension: multi-modality (CLIP), sequences (CPC)
- More Variants

Q&A