Deep Learning for Audio

Lecture 8

Pavel Severilov

Moscow Institute of Physics and Technology

2023

Outline

1. Self-Supervised Learning: introduction
2. Auto-encoding frameworks
3. Siamese architectures
4. Contrastive Representation Learning

Outline

Why Self-Supervised Learning?

- Huge amount of unlabeled data
- Expensive labeling
- Solve many task at once
- Features should be general — not specialized towards a single supervised task

SSL examples from Computer Vision

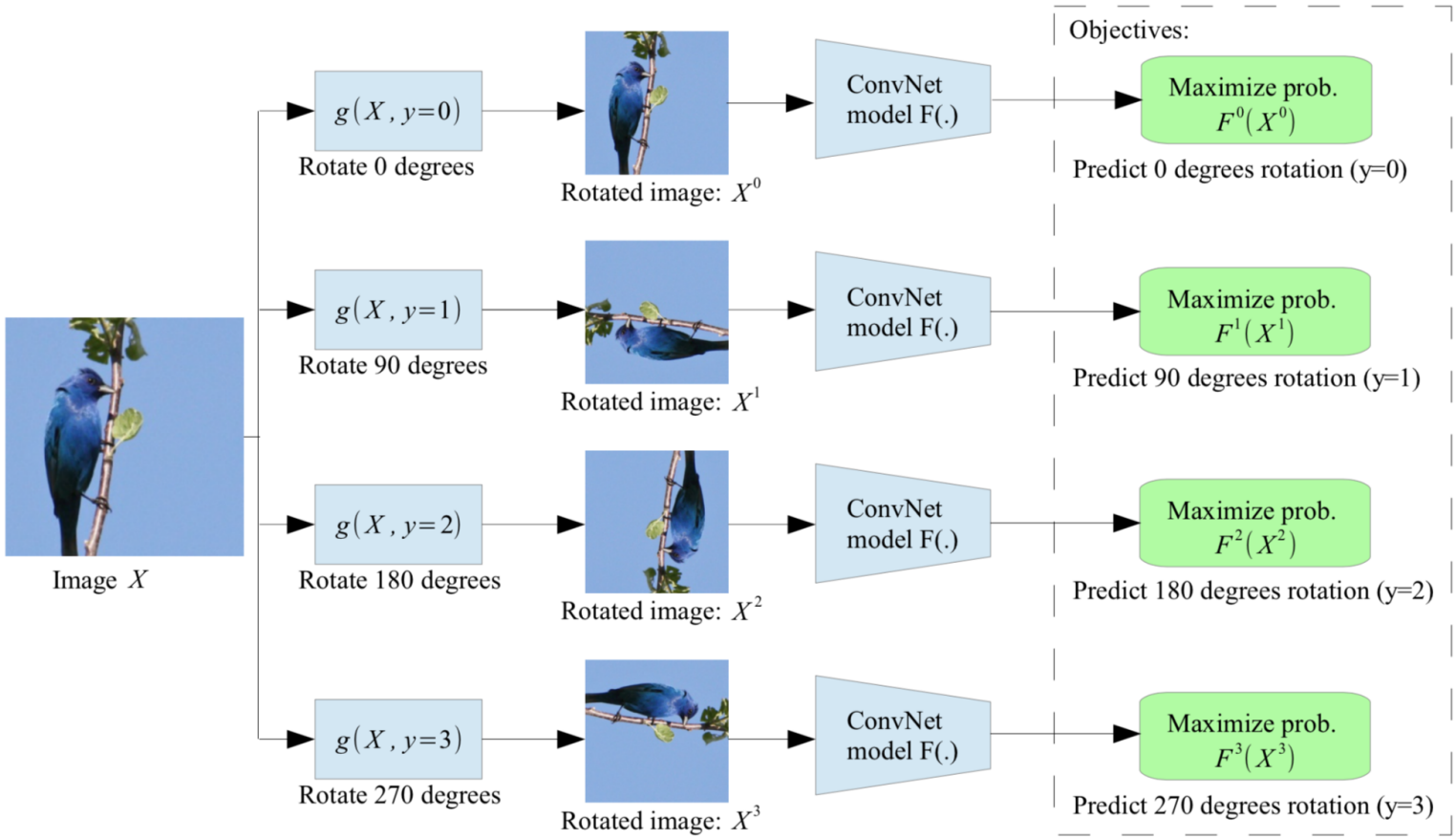


Figure: SSL by rotating the entire input images. The model learns to predict which rotation is applied

SSL examples from Computer Vision

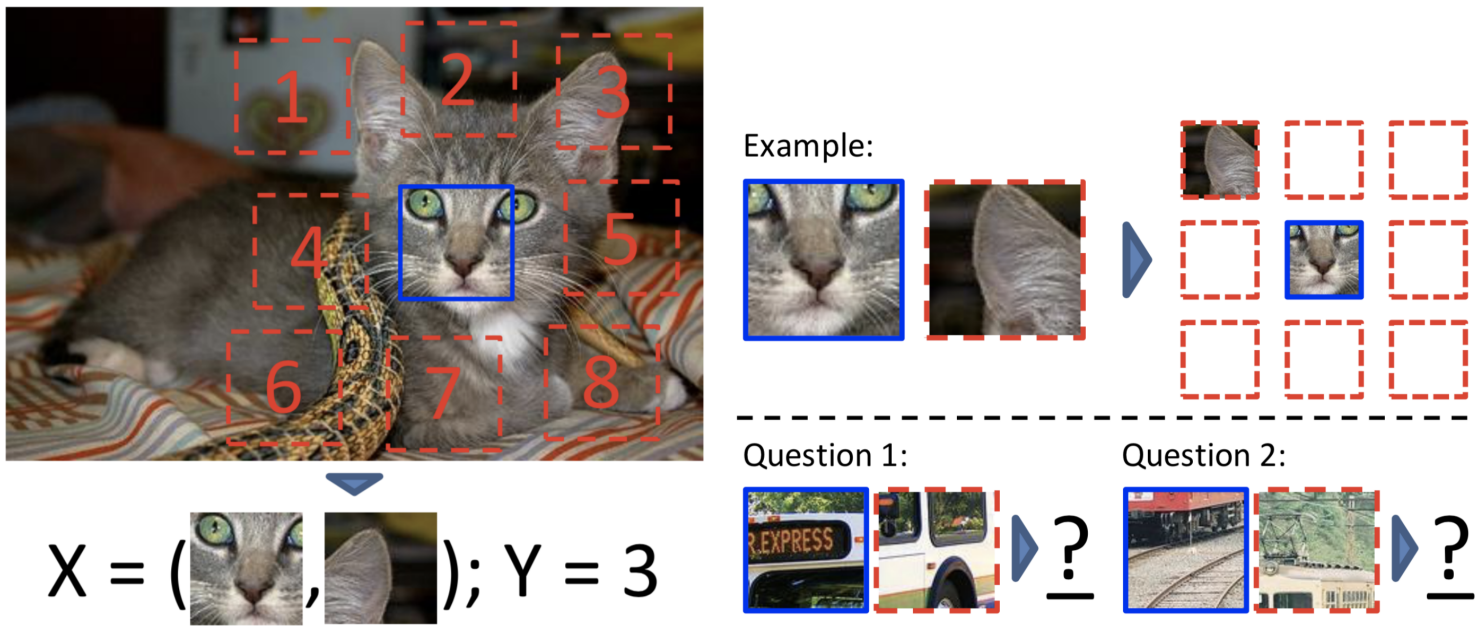
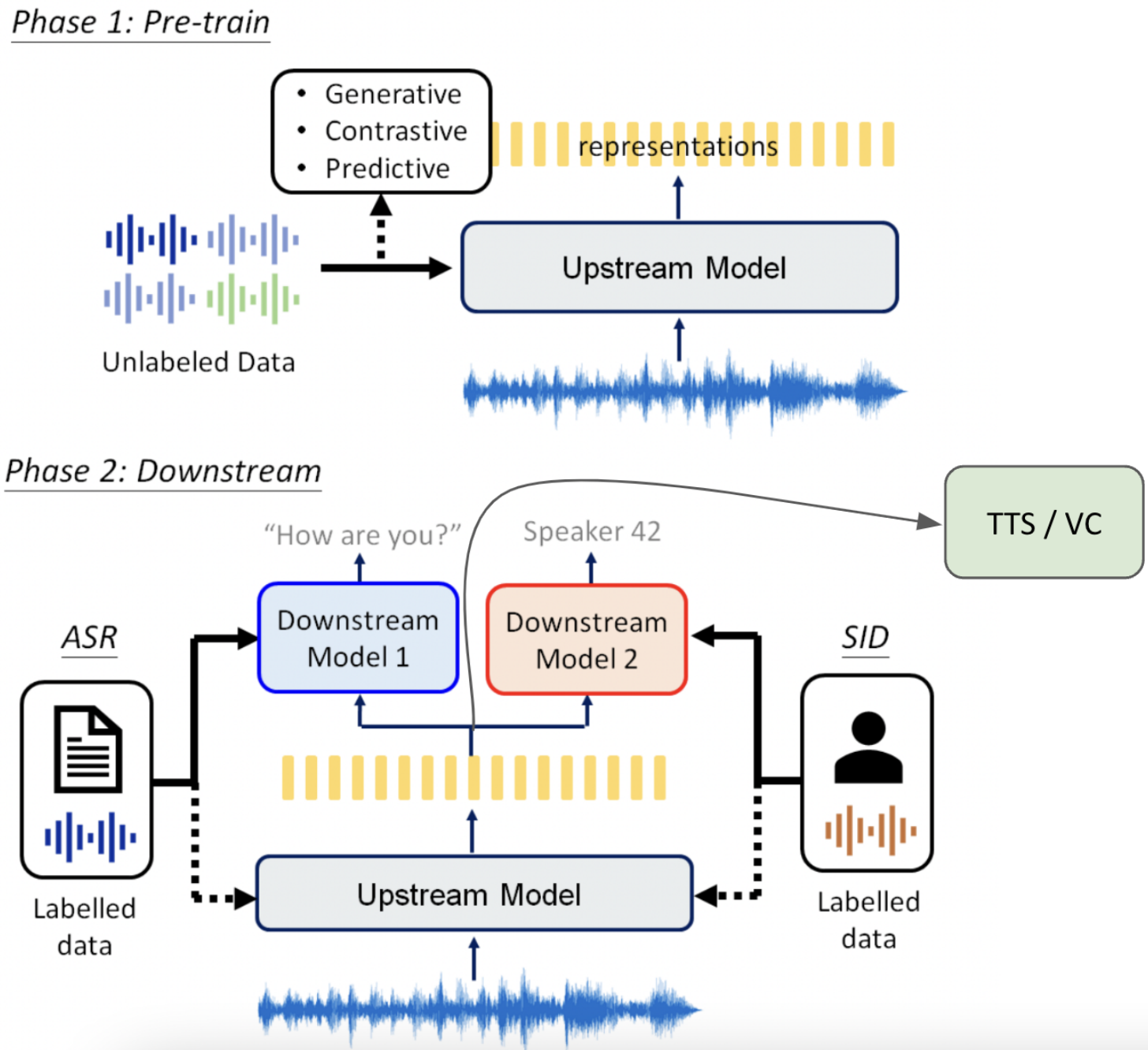
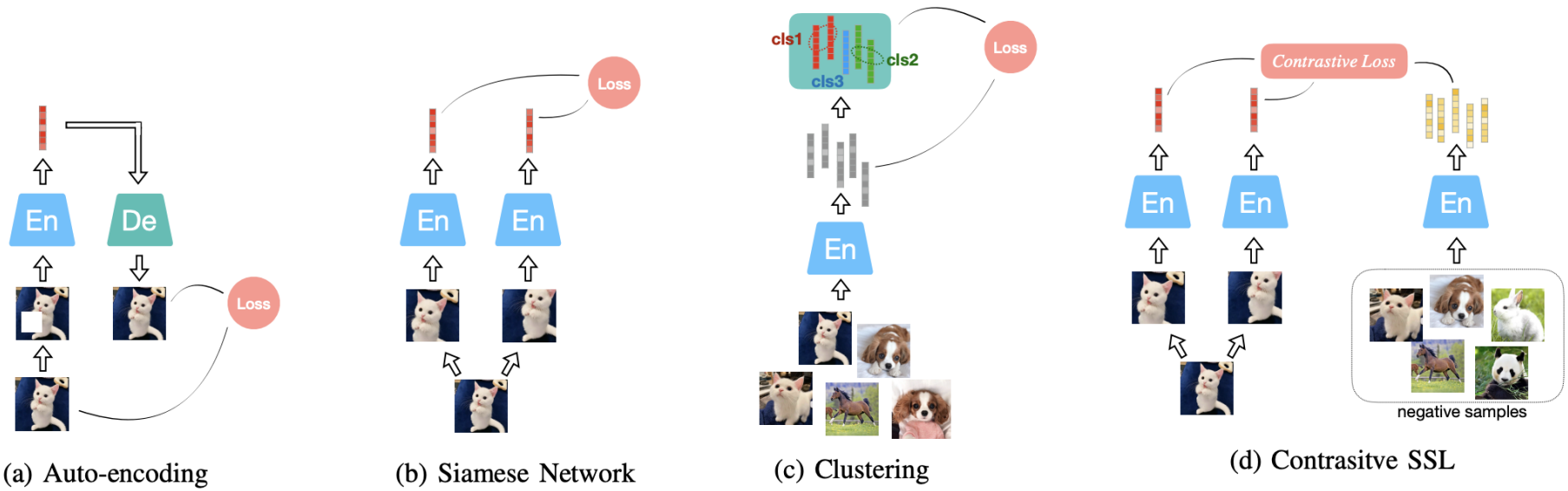


Figure: SSL by predicting the relative position of two random patches.

SSL pipeline



SSL frameworks

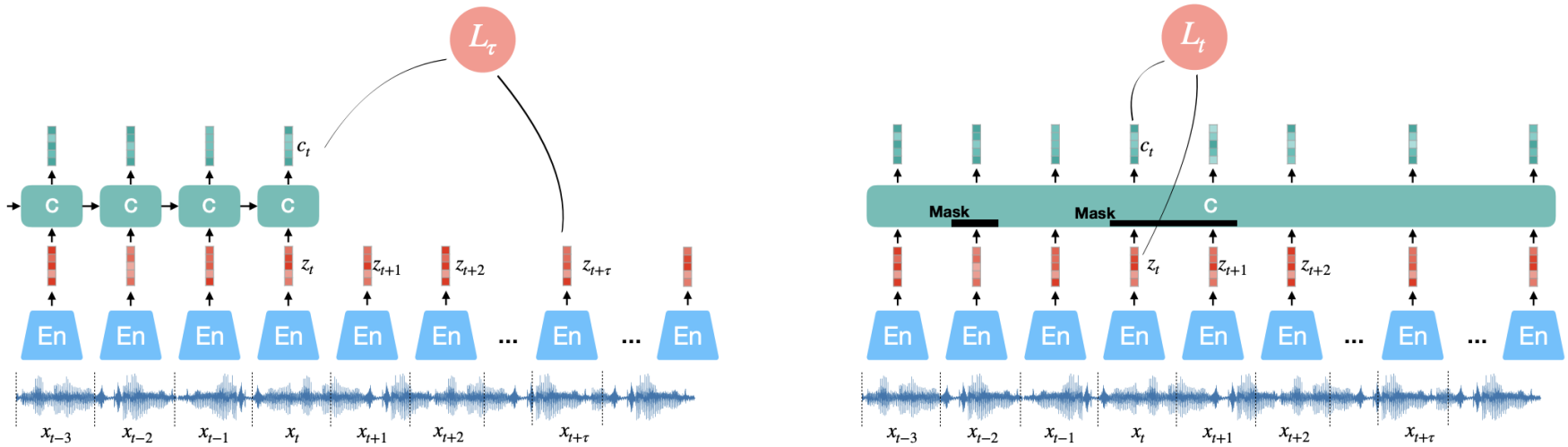


(a) Auto-encoding (b) Siamese Network (c) Clustering (d) Contrasitve SSL

Figure: Predictive SSL frameworks (a-c) and contrastive SSL framework (d)

Outline

APC, MPC



(a) APC

(b) MPC

Figure: Diagrams of auto-regressive predictive coding (APC) and masked predictive coding (MPC).

Outline

Siamese architectures

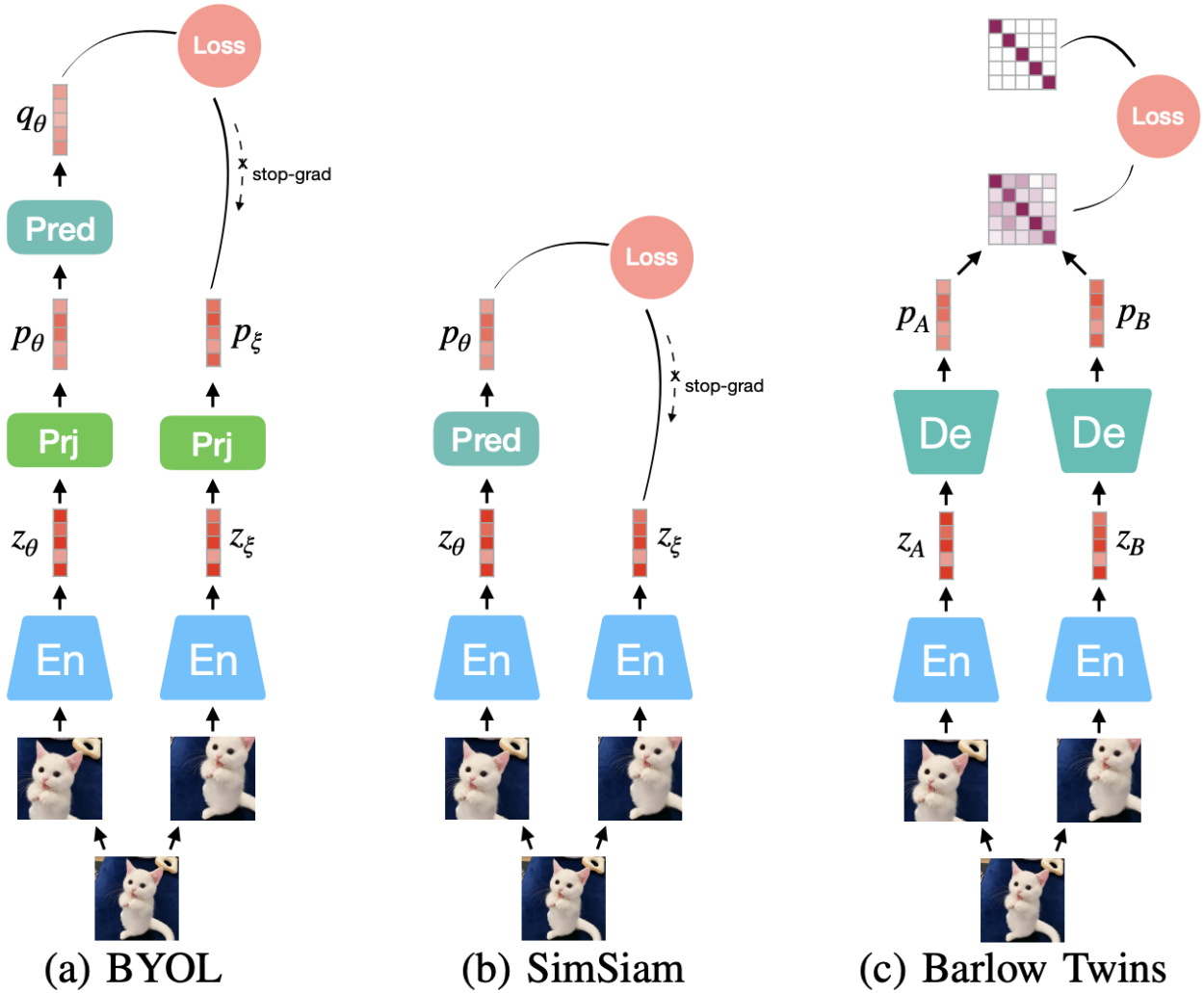


Figure: Diagrams for Predictive Models using Siamese architectures

Outline

Contrastive Loss

- We would like to learn a function $f_\theta(\cdot) : \mathcal{X} \to \mathbb{R}^d$ that encodes samples x_i into an embedding vectors
- Contrastive loss takes a pair of inputs and minimizes the embedding distance when they are from the same class but maximizes the distance otherwise.
-

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

where ϵ – hyperparameter, the lower bound distance between samples of different classes.

Triplet Loss

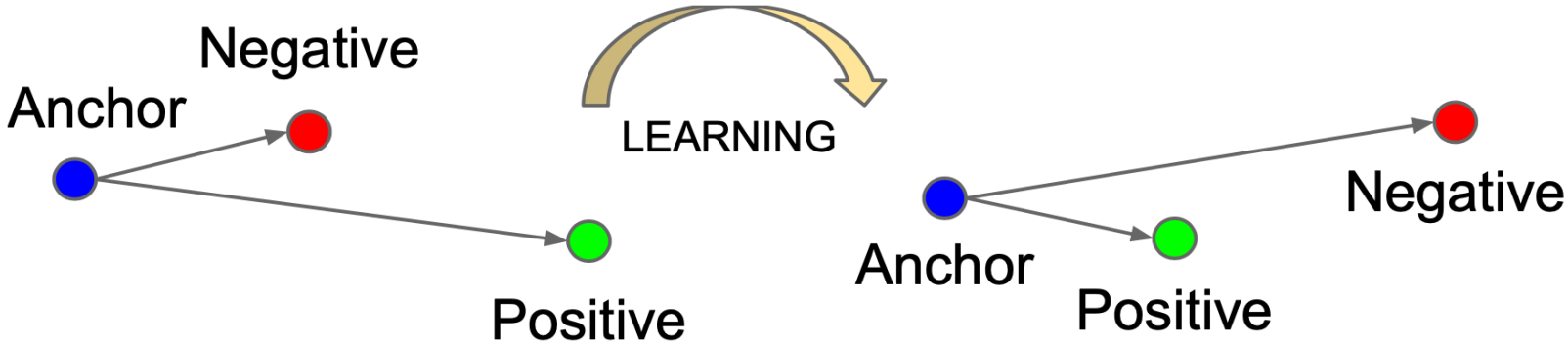


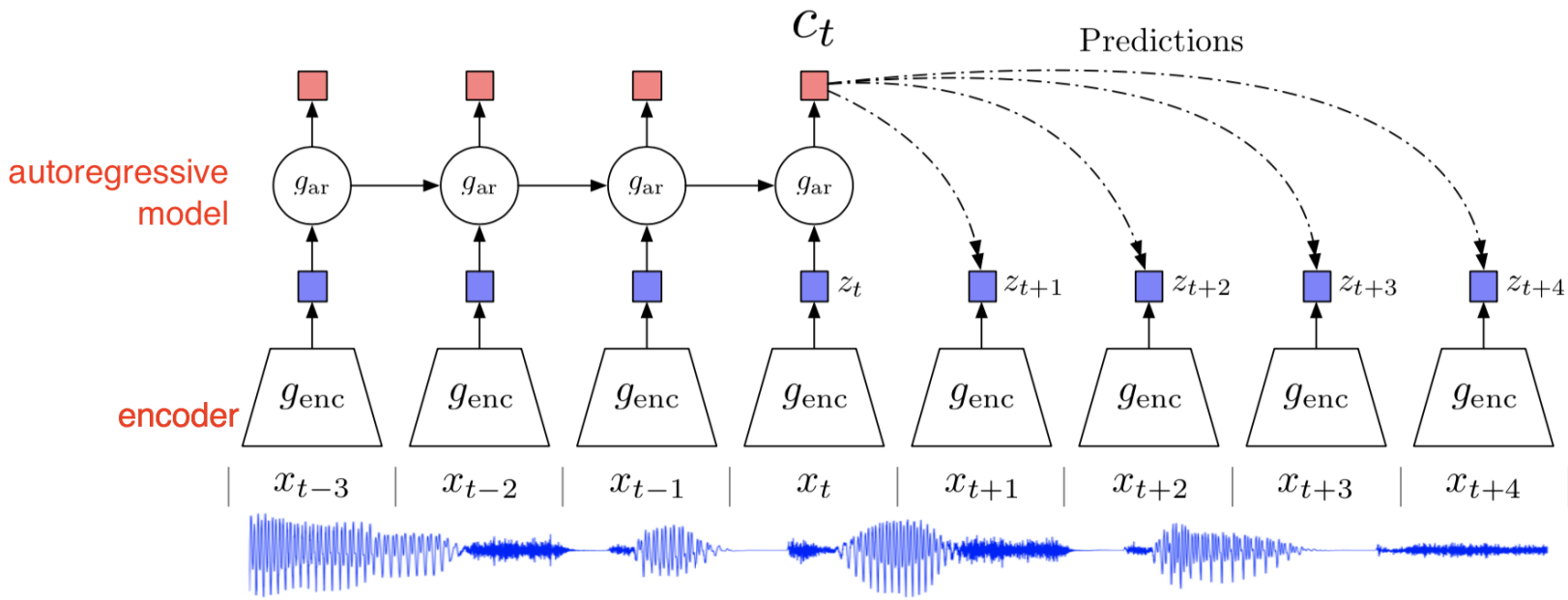
Figure: Illustration of triplet loss given one positive and one negative per anchor.

$$\mathcal{L}_{\text{triplet}}\left(\mathbf{x}, \mathbf{x}^{+}, \mathbf{x}^{-}\right) =$$

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

where ϵ – minimum offset between distances of similar vs dissimilar pairs.

Contrastive Predictive Coding (CPC)



- Non-linear encoder maps input sequence to a sequence of latent representations
- Autoregressive model produces context representation from all history
- Scoring function to maximize mutual information between context and future target
- The end-to-end training relies on the NCE-inspired contrastive loss.

InfoNCE loss

- 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
- InfoNCE loss:

$$\mathcal{L}_{\mathrm{N}} = -\underset{X}{\mathbb{E}}\left[\log \frac{f_k\left(x_{t+k}, c_t\right)}{\sum_{x_j \in X} f_k\left(x_j, c_t\right)}\right]$$

- InfoNCE loss maximize the similarity between a contextualized representation and a localized representation

InfoNCE loss

The InfoNCE loss optimizes the negative log probability of classifying the positive sample correctly

The probability of detecting the positive sample correctly is:

$$p(C = \text{pos} \mid X, \mathbf{c}) = \frac{p\left(x_{\text{pos}} \mid \mathbf{c}\right) \prod_{i=1,\ldots,N; i \neq \text{pos}} p\left(\mathbf{x}_i\right)}{\sum_{j=1}^{N} \left[p\left(\mathbf{x}_j \mid \mathbf{c}\right) \prod_{i=1,\ldots,N; i \neq j} p\left(\mathbf{x}_i\right)\right]} =$$

$$= \frac{\frac{p(\mathbf{x}_{\text{pos}}|c)}{p(\mathbf{x}_{\text{pos}})}}{\sum_{j=1}^{N} \frac{p\left(\mathbf{x}_j|\mathbf{c}\right)}{p\left(\mathbf{x}_j\right)}} = \frac{f\left(\mathbf{x}_{\text{pos}}, \mathbf{c}\right)}{\sum_{j=1}^{N} f\left(\mathbf{x}_j, \mathbf{c}\right)}$$

where the scoring function is $f(\mathbf{x}, \mathbf{c}) \propto \frac{p(\mathbf{x}|\mathbf{c})}{p(\mathbf{x})}$.

Contrastive Predictive Coding (CPC)

Rather than modeling the future observations $p_k\left(x_{t+k} \mid c_t\right)$ directly, CPC models a density function to preserve the mutual information between x_{t+k} and c_t:

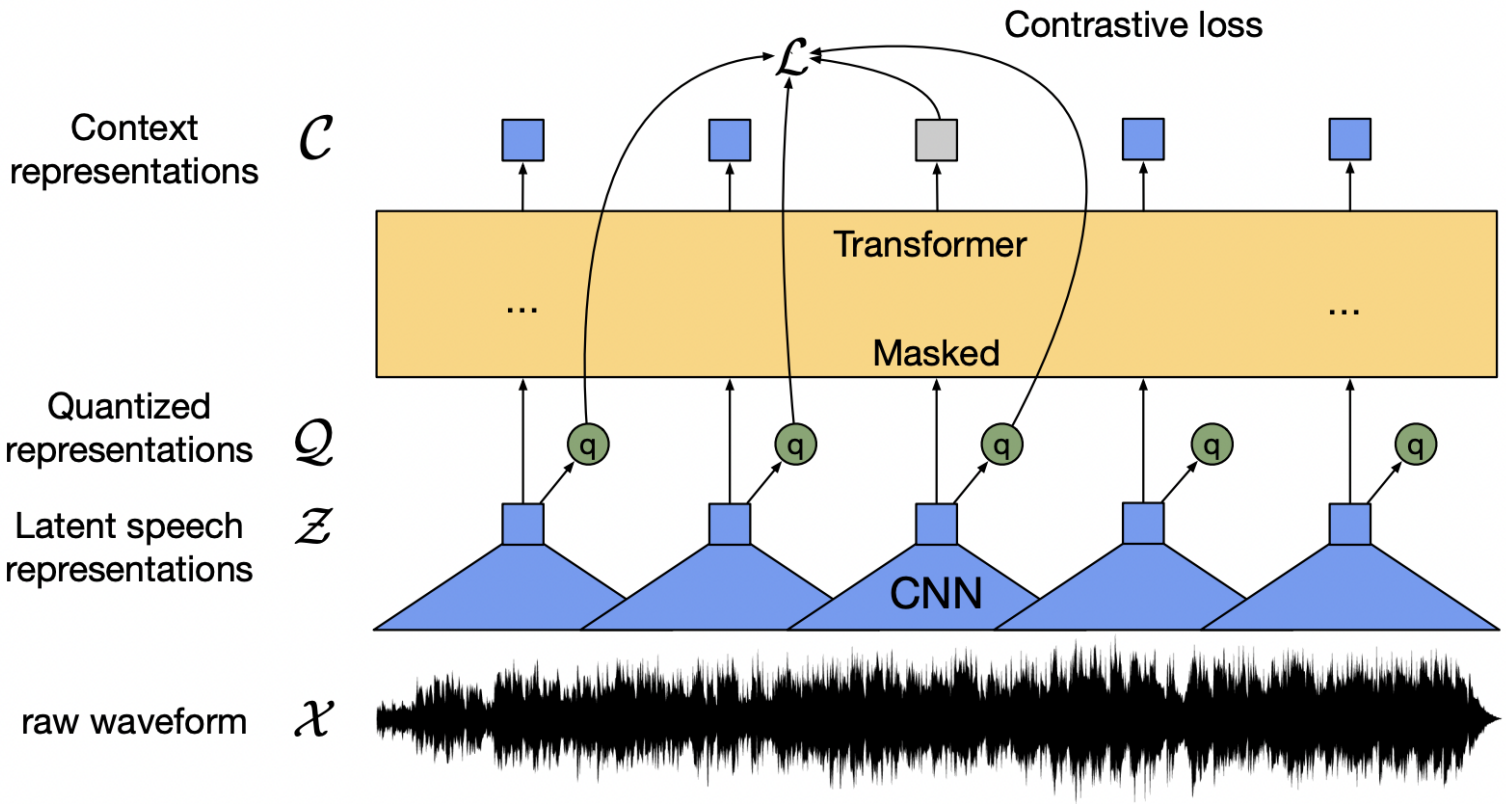
$$f_k\left(x_{t+k}, c_t\right)=\exp\left(z_{t+k}^{\top} W_k c_t\right) \propto \frac{p\left(x_{t+k} \mid c_t\right)}{p\left(x_{t+k}\right)}$$

Mutual Infornation:

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

$$I\left(x_{t+k}, c_t\right) \geq \log(N)-\mathcal{L}_{\mathrm{N}}$$

wav2vec 2.0



- wav2vec (2019): CPC + fully convolutional encoder and autoregressive models + receptive fields
- wav2vec 2.0 (2020): CPC + CNN Encoder + Masking & Transformer