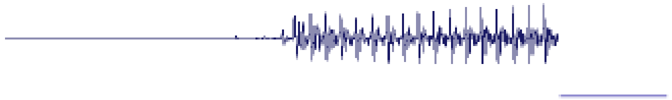
Pretext tasks
4. CPC
|| AR / LPC / VAR / RNN / APC / CPC ||

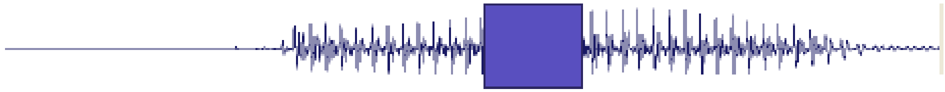
1 2	SELF-PREDICTION	INNATE RELATIONSHIP (Context-based)	1. ROTATION 2. RELATIVE POSITION	IMAGE
3	CONTRASTIVE LEARNING	INTER-SAMPLE CLASSIFICATION	1. Instance Discrimination 2. SimCLR [Contrastive Loss] 3. Theory – Guarantees / Bounds	IMAGE
4	CONTRASTIVE LEARNING	INTER-SAMPLE CLASSIFICATION	Contrastive Predictive Coding (CPC), [NCE, InfoNCE Loss]	AUDIO/ SPEECH
5	SELF-PREDICTION	GENERATIVE (VAE)	1. AE – Variational Bayes 2. VQ-VAE + AR	IMAGE AUDIO/ SPEECH
6	SELF-PREDICTION	GENERATIVE (AR)	1. AR-LM – GPT 2. Masked-LM – BERT	LANGUAGE
7	SELF-PREDICTION	MASKED-GEN (Masked LM for ASR)	1. Wav2Vec / 2.0 2. HuBERT	AUDIO/ SPEECH

Learning with or without supervision – speech and audio

- Next frame prediction



- Masked prediction



- Future prediction
  - To predict future audio features from the historical ones
    - Contrastive predictive coding (CPC) [Oord et al., 2018]
    - Autoregressive predictive coding (APC) [Chung et al., 2019]
    - wav2vec [Schneider et al., 2019]

- [Oord et al., 2018] Representation learning with contrastive predictive coding, arXiv
- [Chung et al., 2019] An unsupervised autoregressive model for speech representation learning, Interspeech
- [Schneider et al., 2019] wav2vec: Unsupervised pre-training for speech recognition, Interspeech

[Oord et al., 2018] Representation learning with **contrastive predictive coding**, arXiv

[Chung et al., 2019] An unsupervised autoregressive model for speech representation learning, Interspeech

[Schneider et al., 2019] wav2vec: Unsupervised pre-training for speech recognition, Interspeech

INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

An Unsupervised Autoregressive Model for Speech Representation Learning

Yu-An Chung, Wei-Ning Hsu, Hao Tang, James Glass

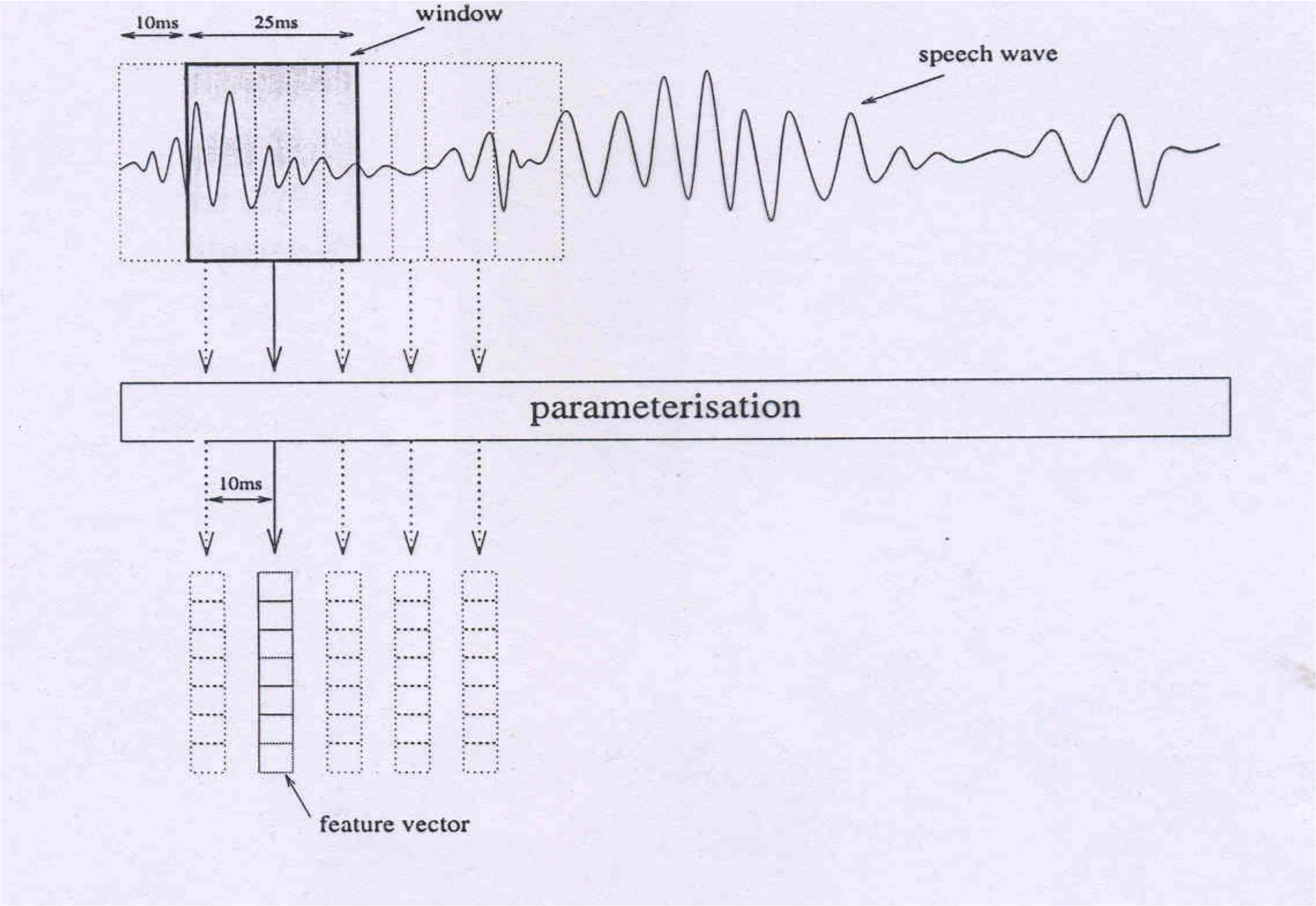
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
`{andyyuan,wnhsu,haotang,glass}@mit.edu`

1380 IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING, VOL. 16, NO. 6, OCTOBER 2022

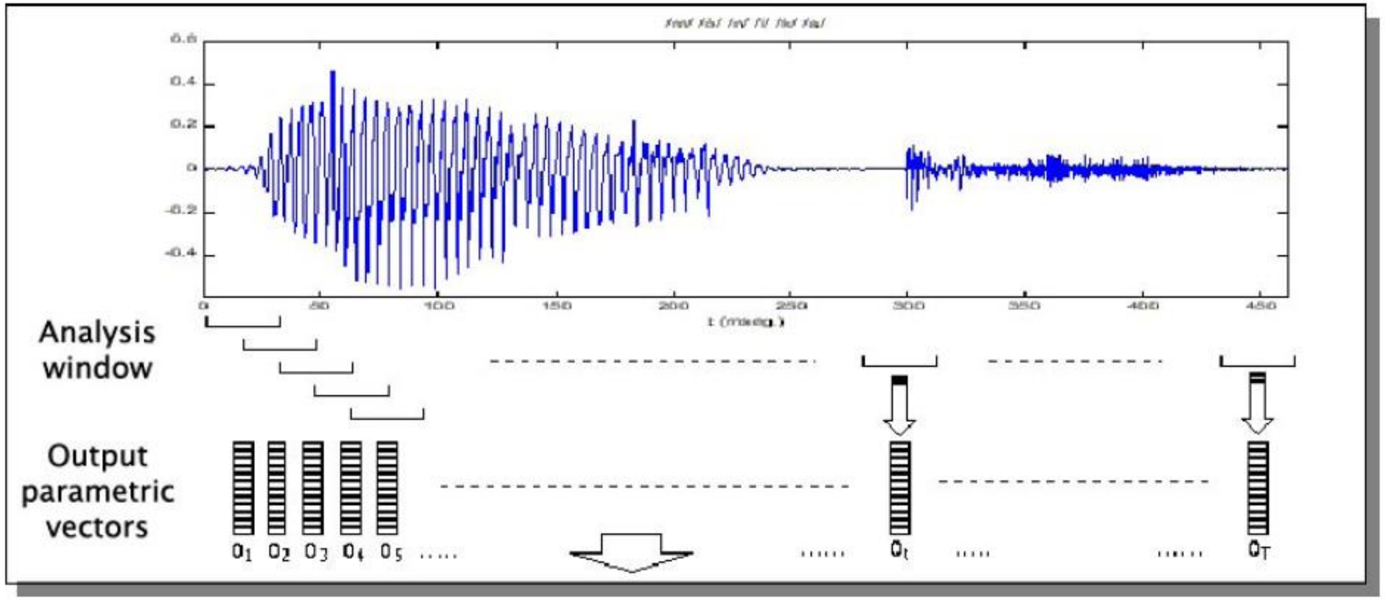
Autoregressive Predictive Coding: A Comprehensive Study

Gene-Ping Yang, Sung-Lin Yeh, Yu-An Chung, James Glass, and Hao Tang

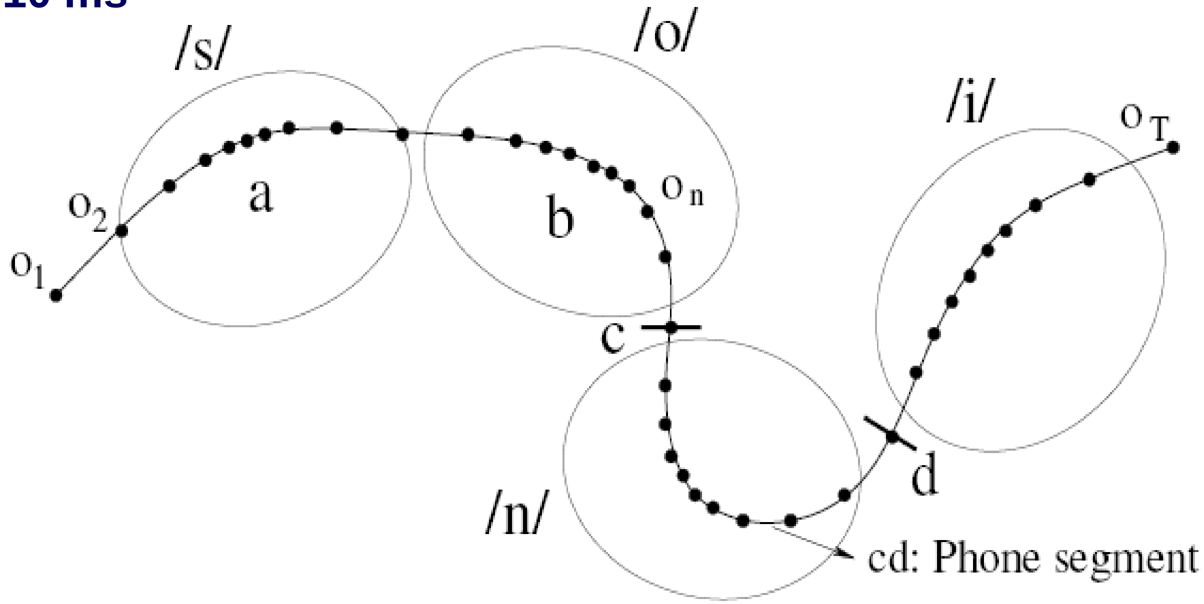
Short-time Analysis and Parameterization



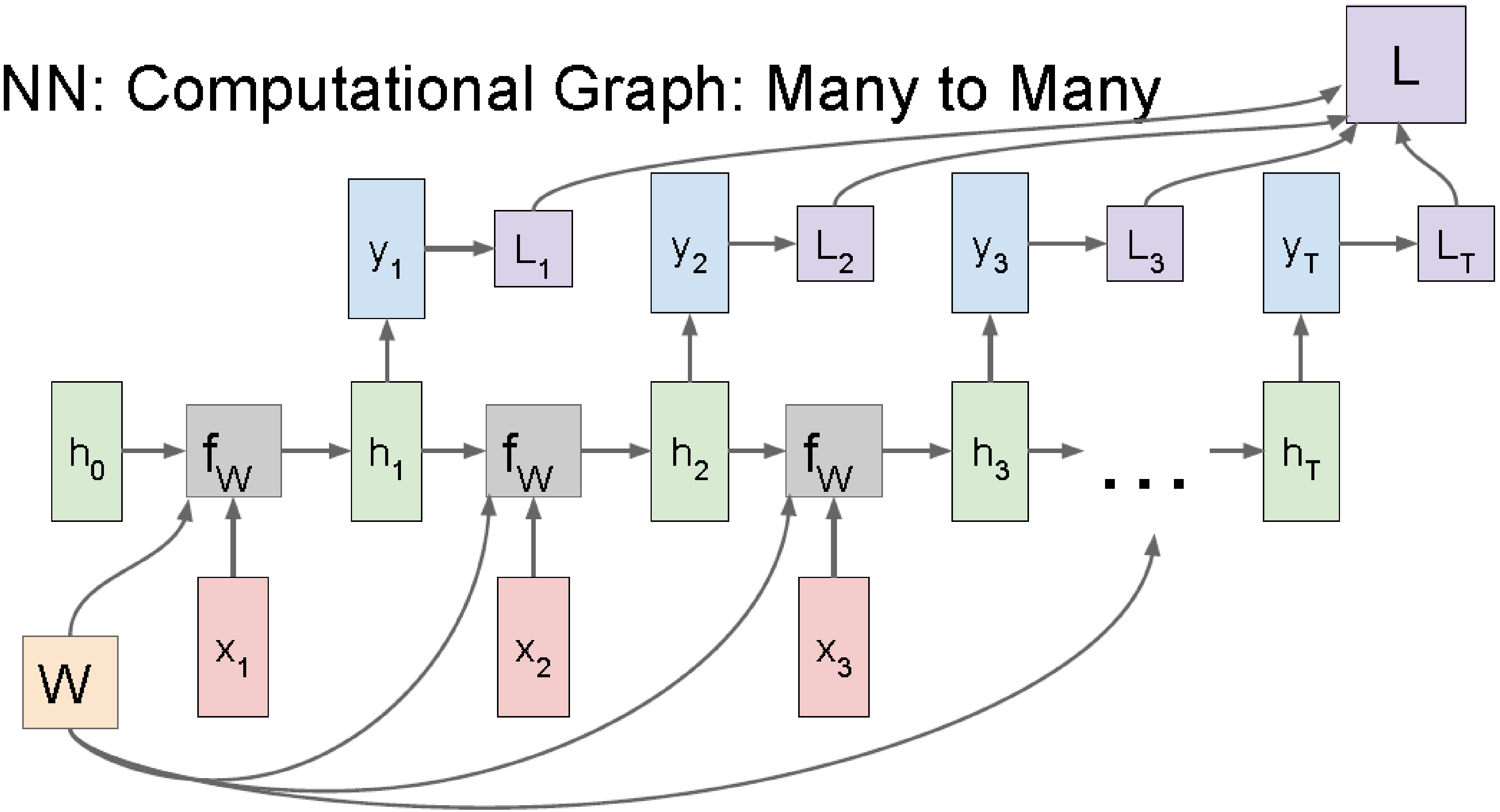
Feature Space



One feature vector every 10 ms

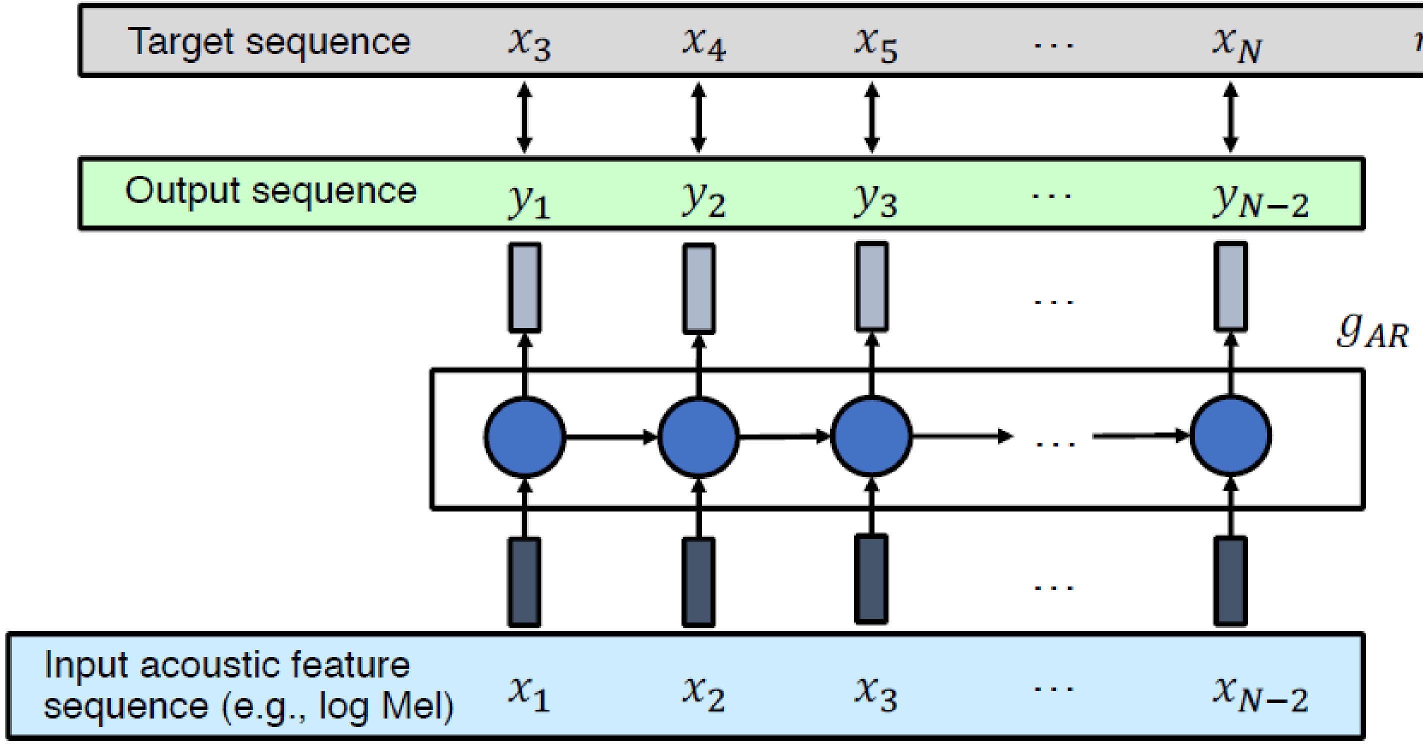


RNN: Computational Graph: Many to Many



Autoregressive Predictive Coding (APC)

- Given a previous context $(\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_t})$, APC tries to predict a future audio feature $\boldsymbol{x_{t+n}}$ that is $\boldsymbol{n}$ steps ahead of $\boldsymbol{x_t}$
  - Uses an autoregressive model g_{AR} to summarize history and produce output
  - $n \geq 1$ encourages g_{AR} to infer more global underlying structures of the data rather than simply exploiting local smoothness of speech signals



Training

$$\underset{\{g_{AR}, W\}}{\operatorname{argmin}} \ \sum_{t=1}^{N-n} |x_{t+n} - y_t|,$$

$$y_t = g_{AR}(x_1, \dots, x_t) \cdot W$$

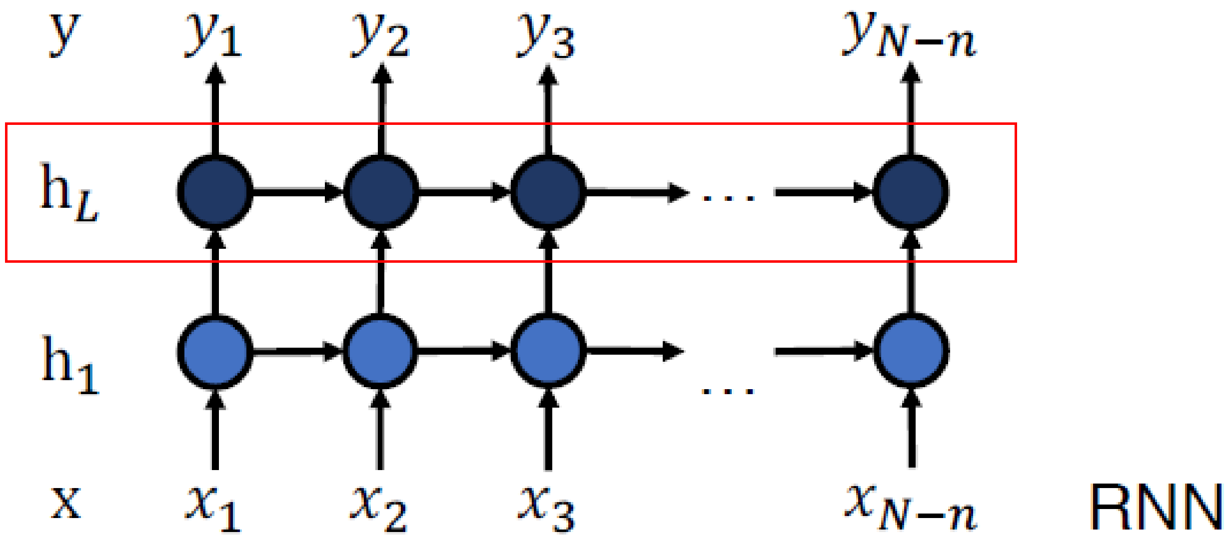
W is a linear transformation that maps g_{AR}'s output back to x_t's dimensionality

Types of autoregressive model g_{AR}

- g_{AR}
  - Input: $\mathbf{x} = (\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_N})$
  - Output: $\mathbf{y} = (\boldsymbol{y_1}, \boldsymbol{y_2}, \dots, \boldsymbol{y_N})$

- L-layer Unidirectional RNN:

$$\mathrm{h}_0 = \mathrm{x}$$
$$\mathrm{h}_l = \mathrm{RNN}^{(l)}(h_{l-1}), \forall l \in [1, L]$$
$$\mathrm{y} = \mathrm{h}_L \cdot W$$



- Feature extraction: $\mathbf{h}_L$

Representation Learning with Contrastive Predictive Coding

Aaron van den Oord
DeepMind
avdnoord@google.com

Yazhe Li
DeepMind
yazhe@google.com

Oriol Vinyals
DeepMind
vinyals@google.com

CPC

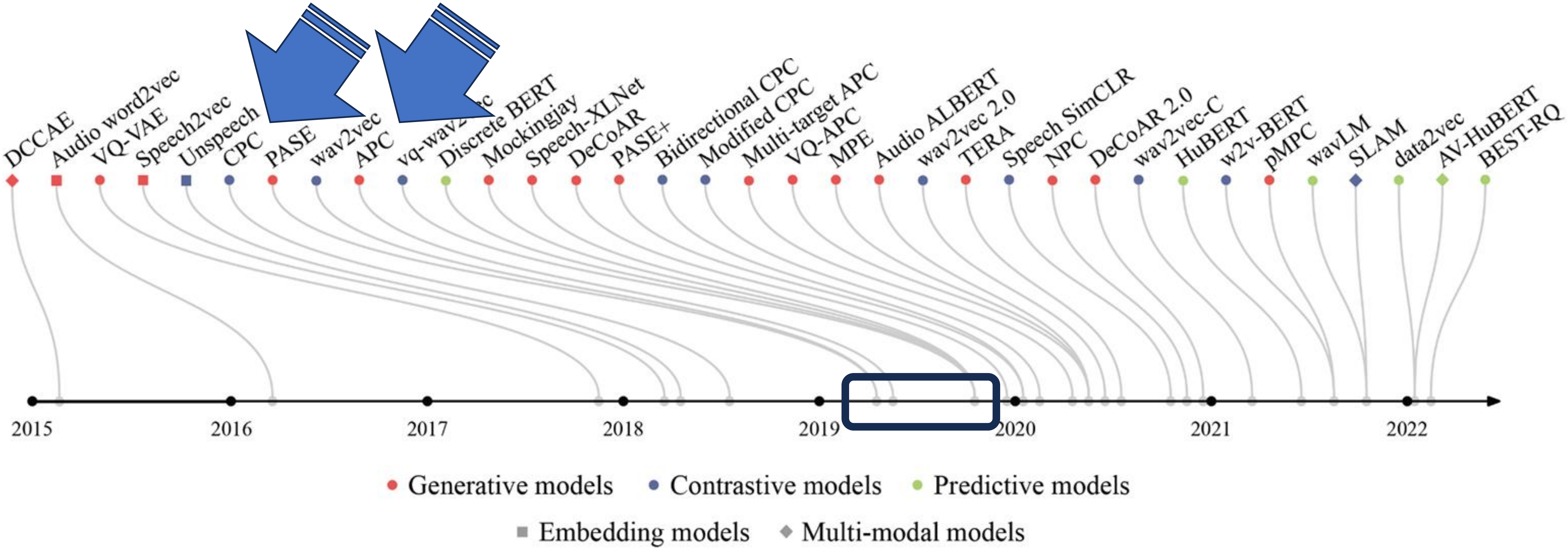
- The first successful representation learning approach for speech data.
- It triggered lots of research in speech representation learning.

CPC: The pretext task

- Distinguish correct (positive) samples from wrong (negative) ones.
- **But, how do we choose positive and negative examples?**

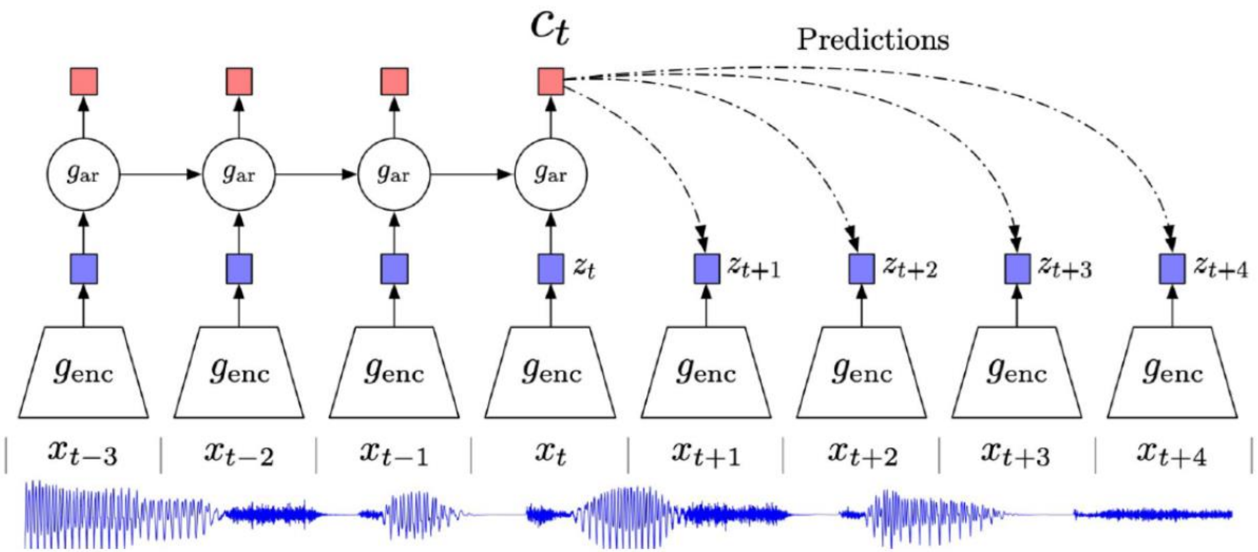
van den Oord et al, 2019 “Representation Learning with Contrastive Predictive Coding”

Speech representation learning methods



CPC

CPC example: modeling audio sequences



Contrastive: contrast between "right" and "wrong" sequences using contrastive learning.

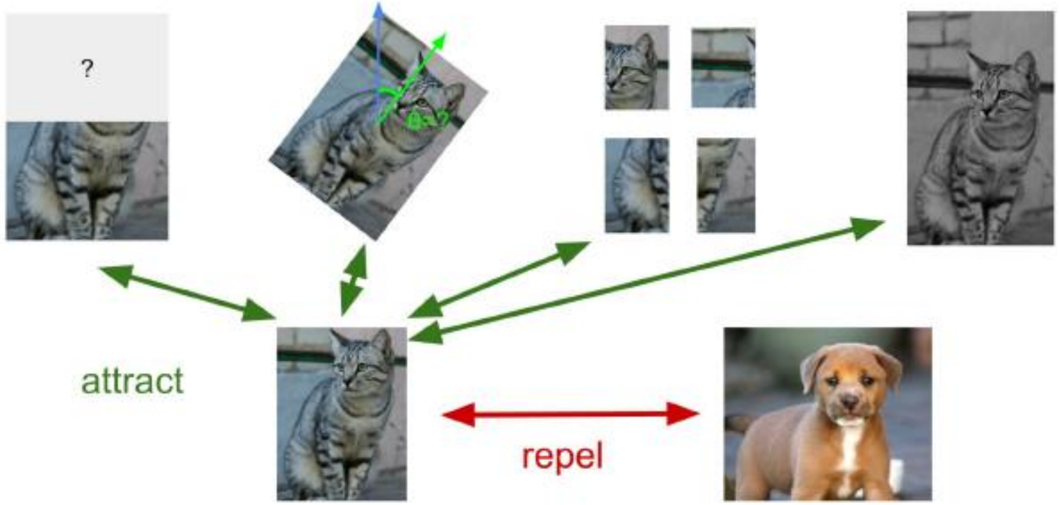
Predictive: the model has to predict future patterns given the current context.

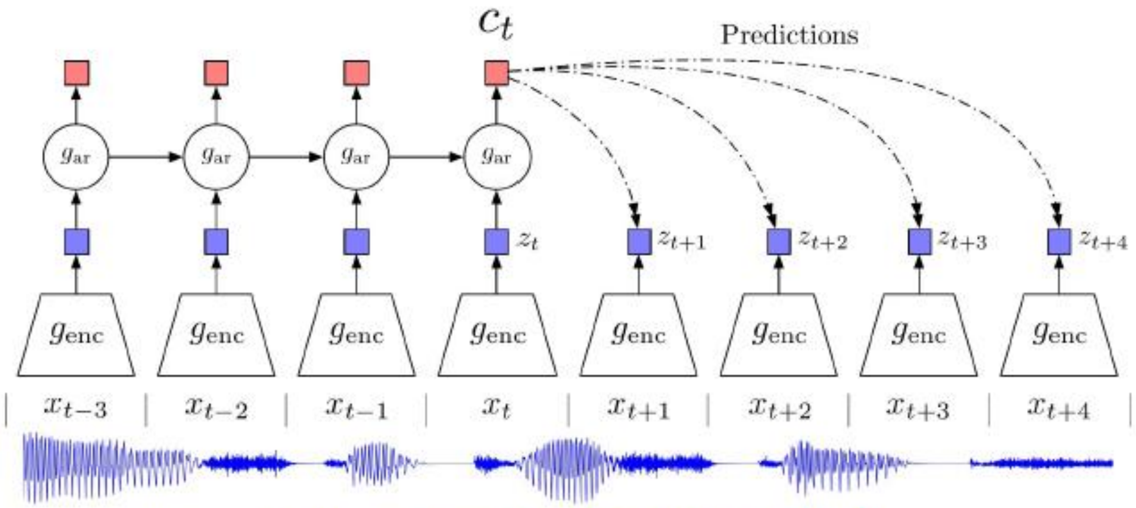
Coding: the model learns useful feature vectors, or "code", for downstream tasks, similar to other self-supervised methods.

Contrastive representation learning

- Intuition and formulation
- Instance contrastive learning: SimCLR and MOCO
- Sequence contrastive learning: CPC

Instance vs. Sequence Contrastive Learning



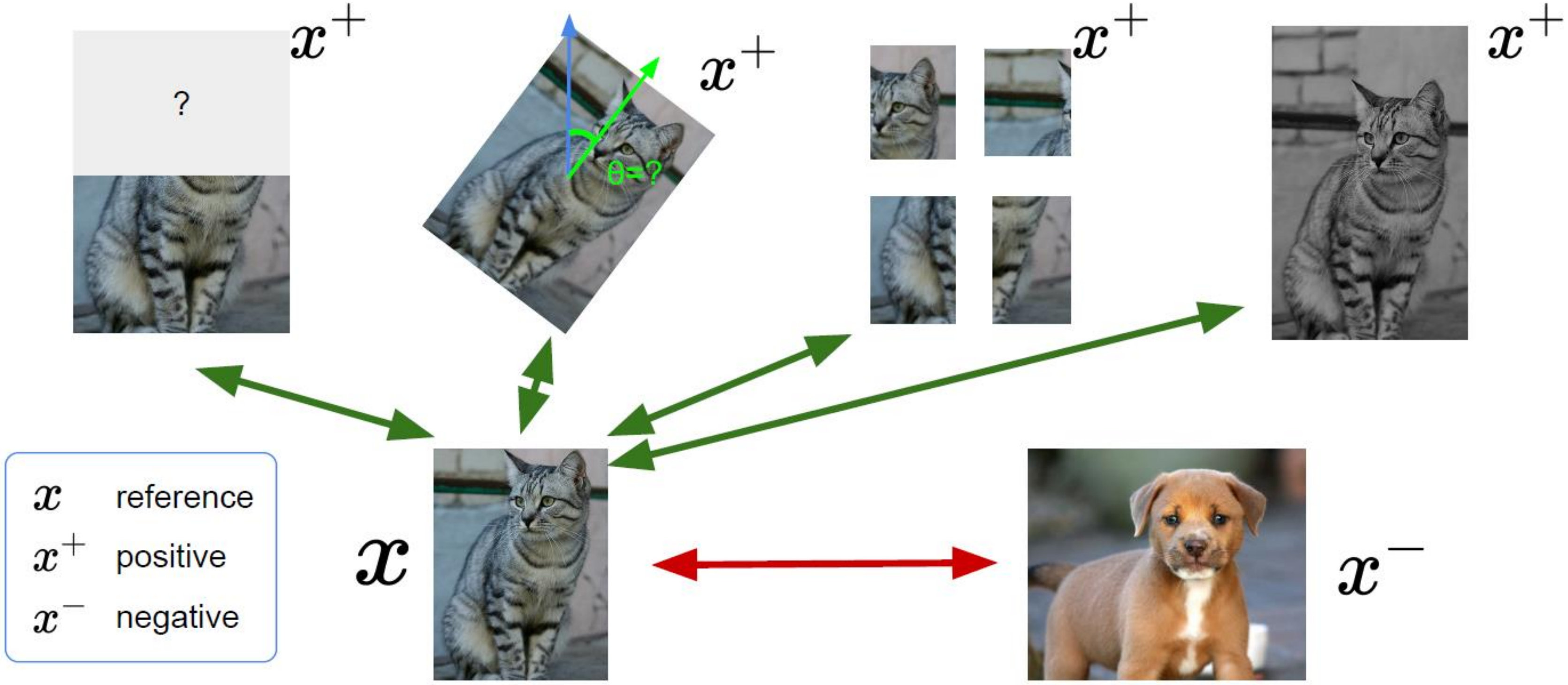


Source: van den Oord et al., 2018

Instance-level contrastive learning:
contrastive learning based on
positive & negative instances.
Examples: SimCLR, MoCo

Sequence-level contrastive learning:
contrastive learning based on
sequential / temporal orders.
Example: **Contrastive Predictive Coding (CPC)**

Contrastive Representation Learning



A formulation of contrastive learning

What we want:

$$\text{score}(f(x), f(x^+)) >> \text{score}(f(x), f(x^-))$$

x: reference sample; x^+ positive sample; x^- negative sample

Given a chosen score function, we aim to learn an **encoder function** *f* that yields high score for positive pairs (x, x^+) and low scores for negative pairs (x, x^-).

A formulation of contrastive learning

Loss function given 1 positive sample and N - 1 negative samples:

$$L = -\mathbb{E}_X \left[\log \frac{\exp(s(f(x), f(x^+)))}{\exp(s(f(x), f(x^+))) + \sum_{j=1}^{N-1} \exp(s(f(x), f(x_j^-)))} \right]$$

x

x^+

x

x_1^-

x_2^-

x_3^-

...

A formulation of contrastive learning

Loss function given 1 positive sample and N - 1 negative samples:

$$L = -\mathbb{E}_X \left[\log \frac{\exp(s(f(x), f(x^+)))}{\exp(s(f(x), f(x^+))) + \sum_{j=1}^{N-1} \exp(s(f(x), f(x_j^-)))} \right]$$

score for the positive pair

score for the N-1 negative pairs

This seems familiar …

Cross entropy loss for a N-way softmax classifier!

I.e., learn to find the positive sample from the N samples

A formulation of contrastive learning

Loss function given 1 positive sample and N - 1 negative samples:

$$L = -\mathbb{E}_X \left[\log \frac{\exp(s(f(x), f(x^+)))}{\exp(s(f(x), f(x^+))) + \sum_{j=1}^{N-1} \exp(s(f(x), f(x_j^-)))} \right]$$

Commonly known as the InfoNCE loss (van den Oord et al., 2018)

A *lower bound* on the mutual information between *f(x)* and *f(x^+)*

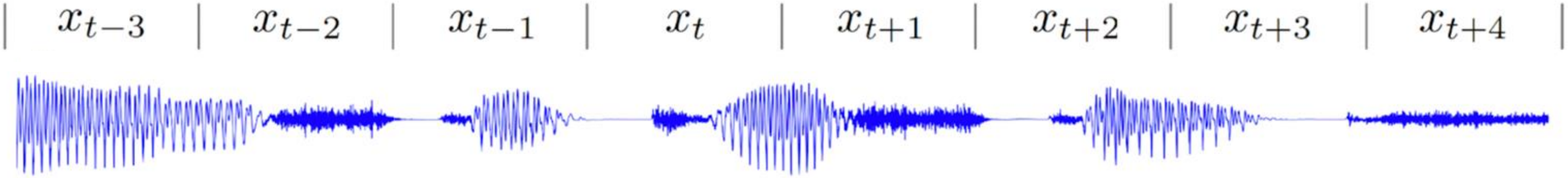
$$MI[f(x), f(x^+)] - \log(N) \geq -L$$

The larger the negative sample size (*N*), the tighter the bound

CPC: The pretext task

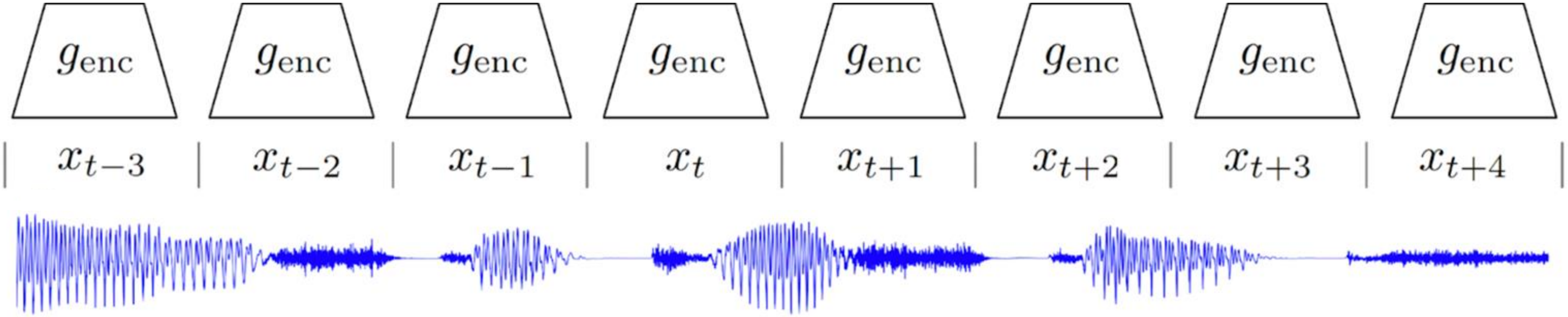
van den Oord et al, 2019 “Representation Learning with Contrastive Predictive Coding”

CPC: The pretext task



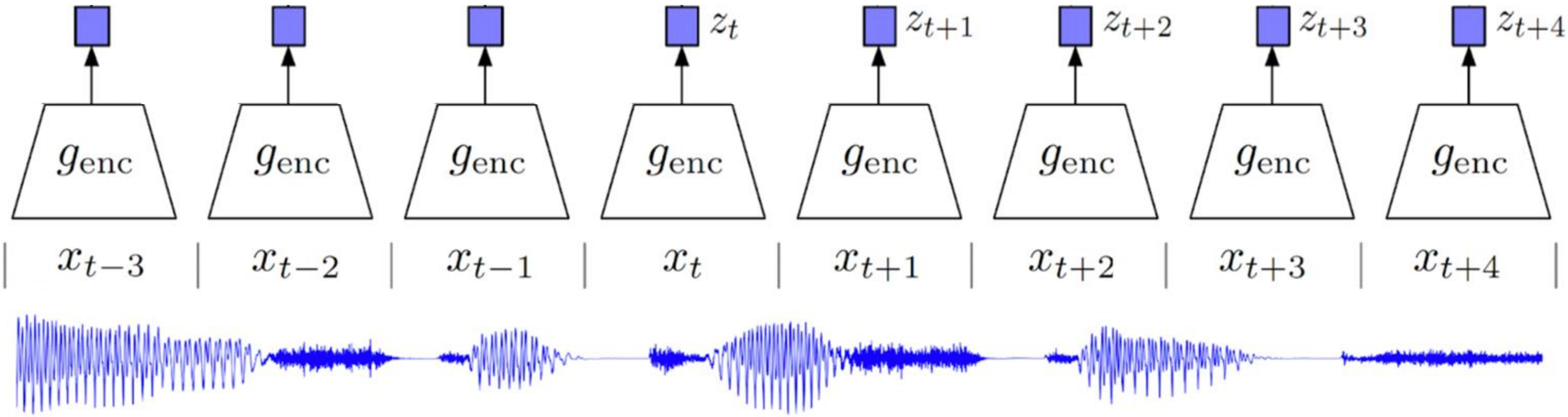
van den Oord et al, 2019 “Representation Learning with Contrastive Predictive Coding”

CPC: The pretext task



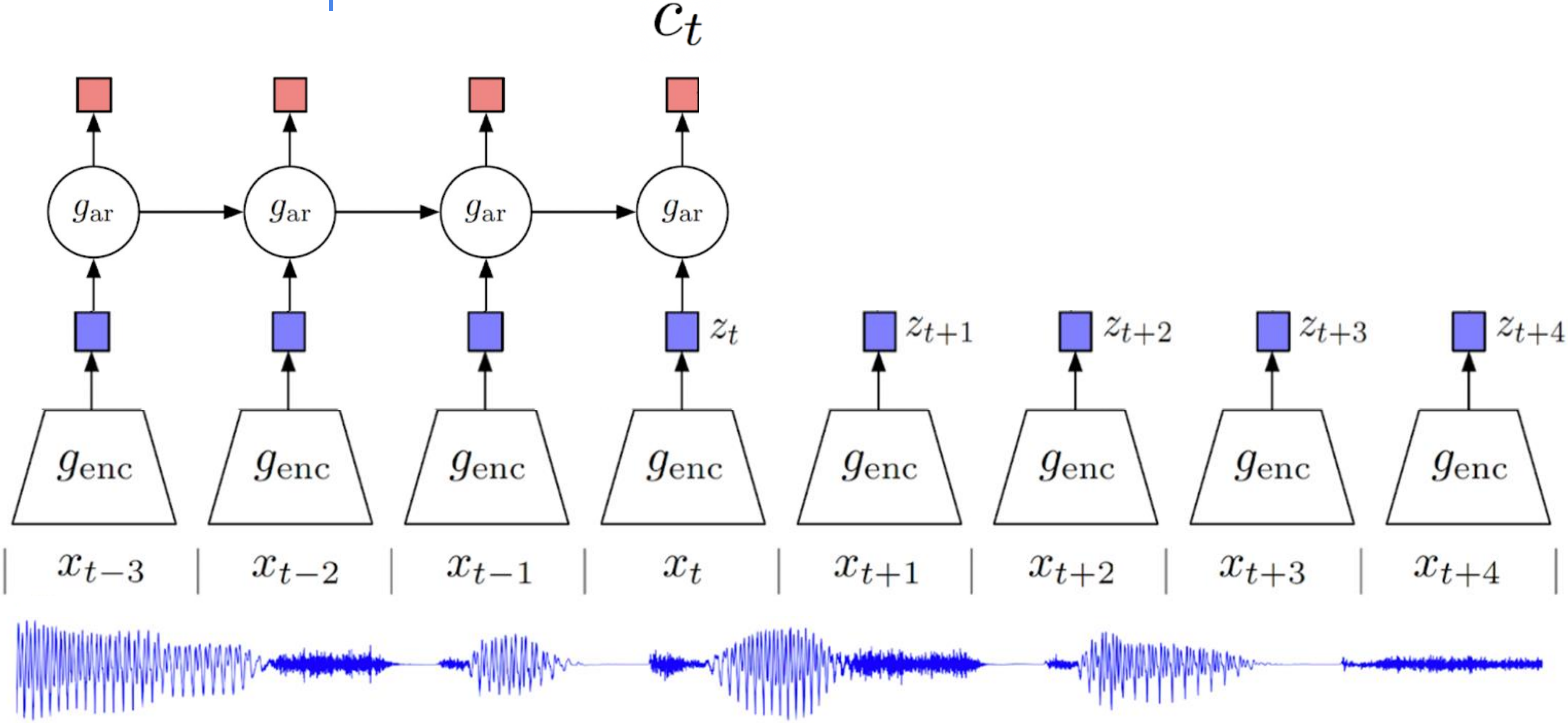
van den Oord et al, 2019 "Representation Learning with Contrastive Predictive Coding"

CPC: The pretext task



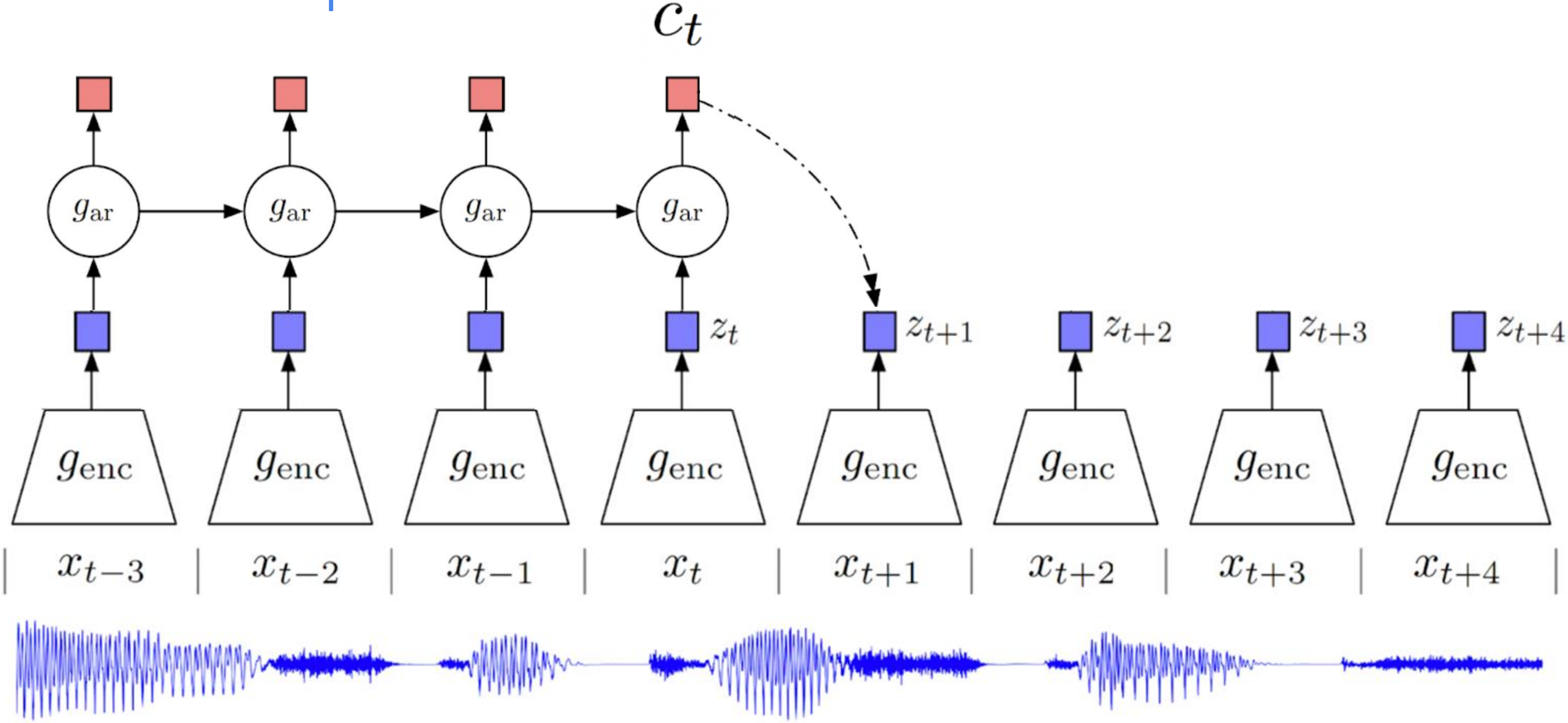
van den Oord et al, 2019 “Representation Learning with Contrastive Predictive Coding”

CPC: The pretext task



van den Oord et al, 2019 “Representation Learning with Contrastive Predictive Coding”

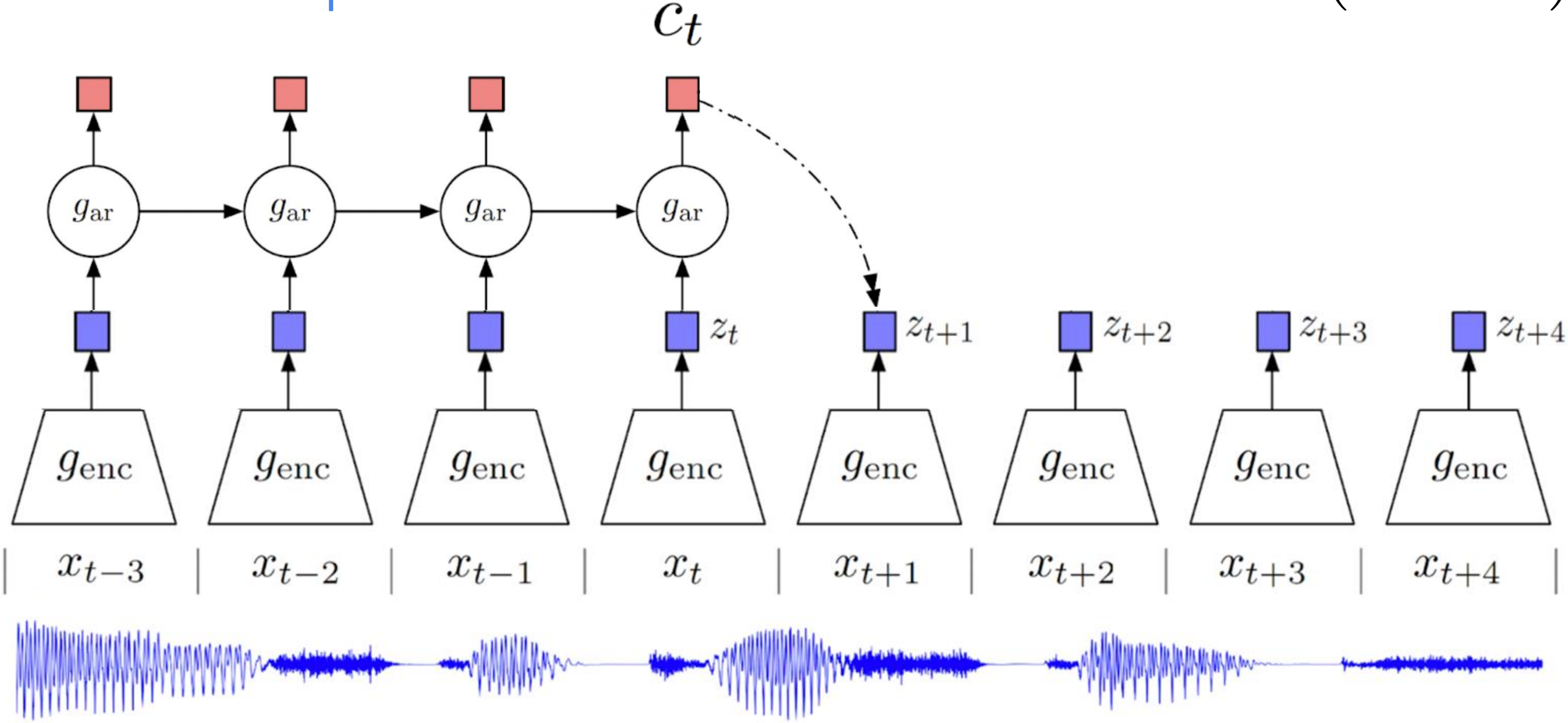
CPC: The pretext task



van den Oord et al, 2019 “Representation Learning with Contrastive Predictive Coding”

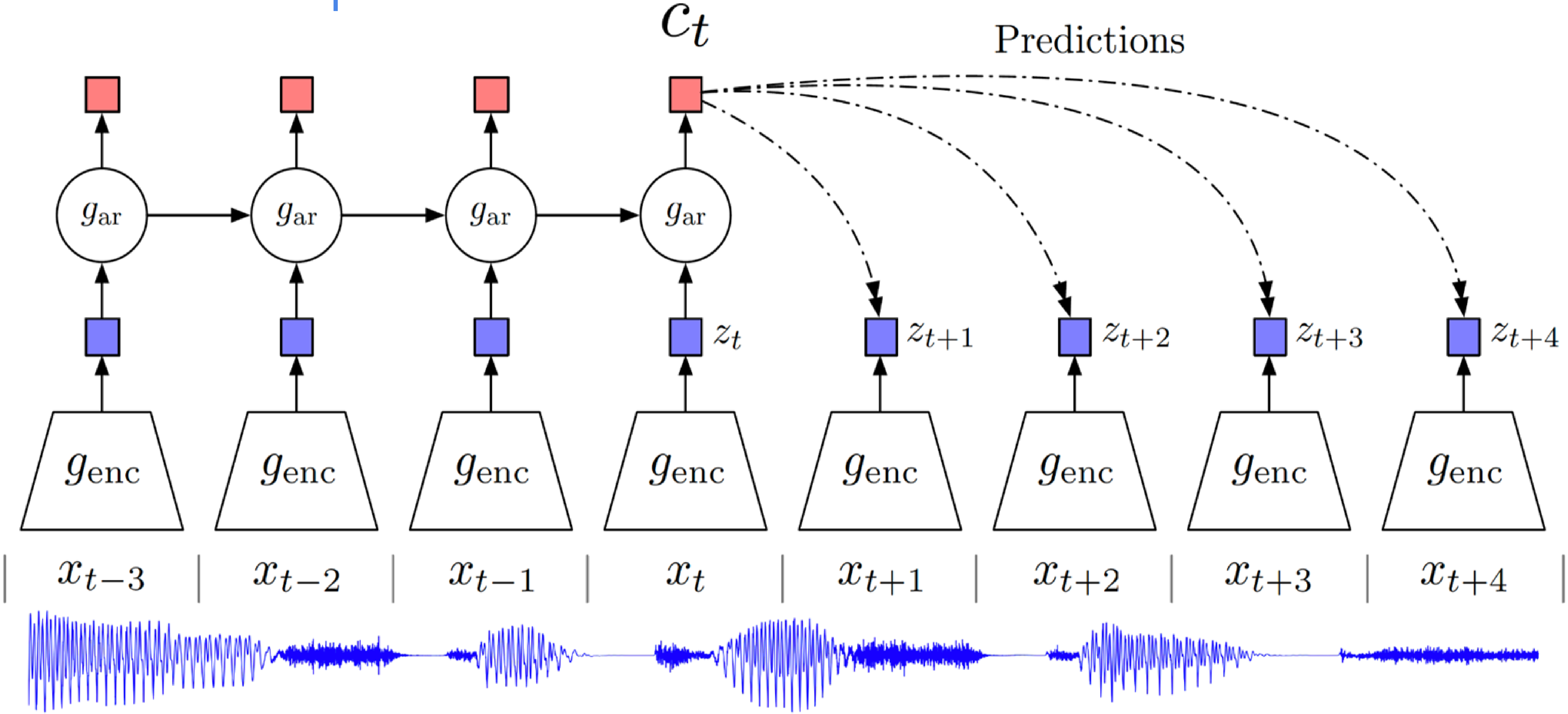
CPC: The pretext task

$$f_k(x_{t+k}, c_t) = \exp\left(z_{t+k}^T W_k c_t\right)$$



van den Oord et al, 2019 “Representation Learning with Contrastive Predictive Coding”

CPC: The pretext task



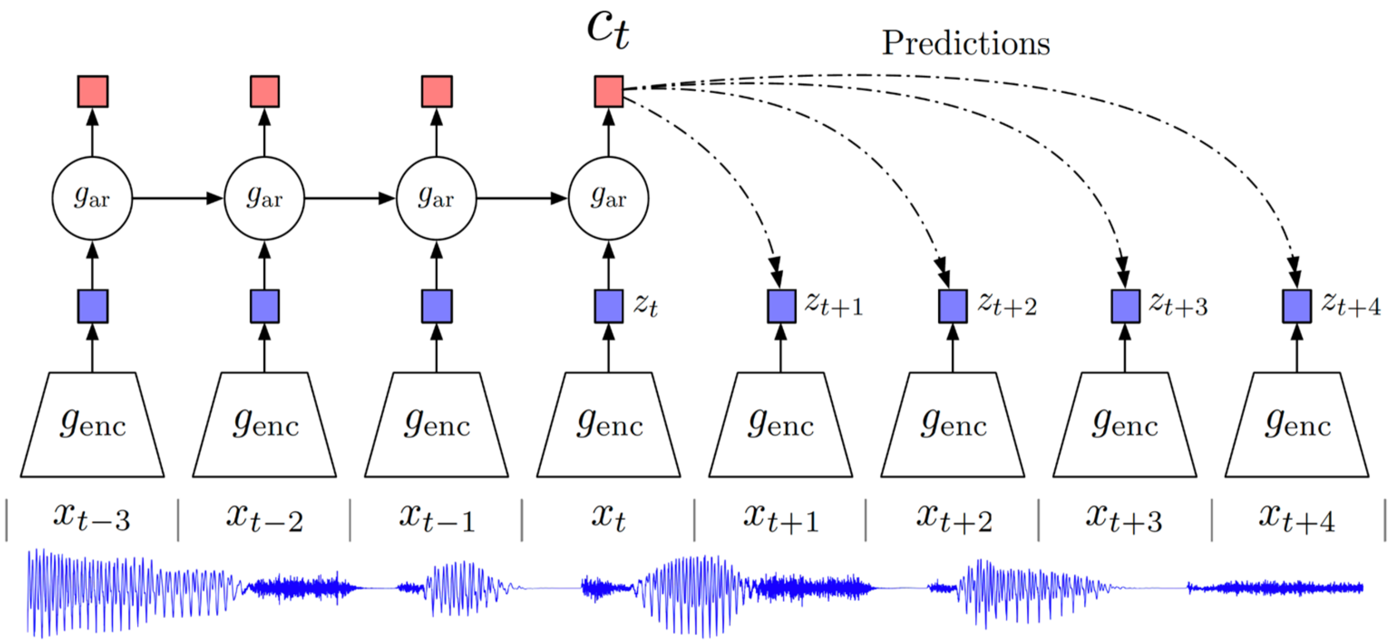
van den Oord et al, 2019 “Representation Learning with Contrastive Predictive Coding”

CPC: The pretext task

$$f_k(x_{t+k}, c_t) = \exp\left(z_{t+k}^T W_k c_t\right)$$

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

- InfoNCE maximizes the mutual information between the input signal and the learned latent variables C.
- Strategies for sampling negative and positive examples determine the nature of representations, e.g., whether they are good for ASR or Speaker ID.



van den Oord et al, 2019 “Representation Learning with Contrastive Predictive Coding”

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

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

Linear classifier trained on top of features.

On using non-linear classifier, CPC accuracy increases to 72.5—not all information is linearly accessible.

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

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

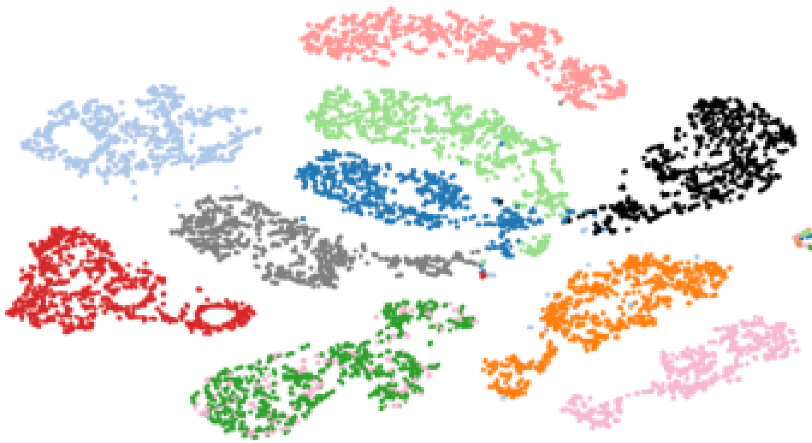
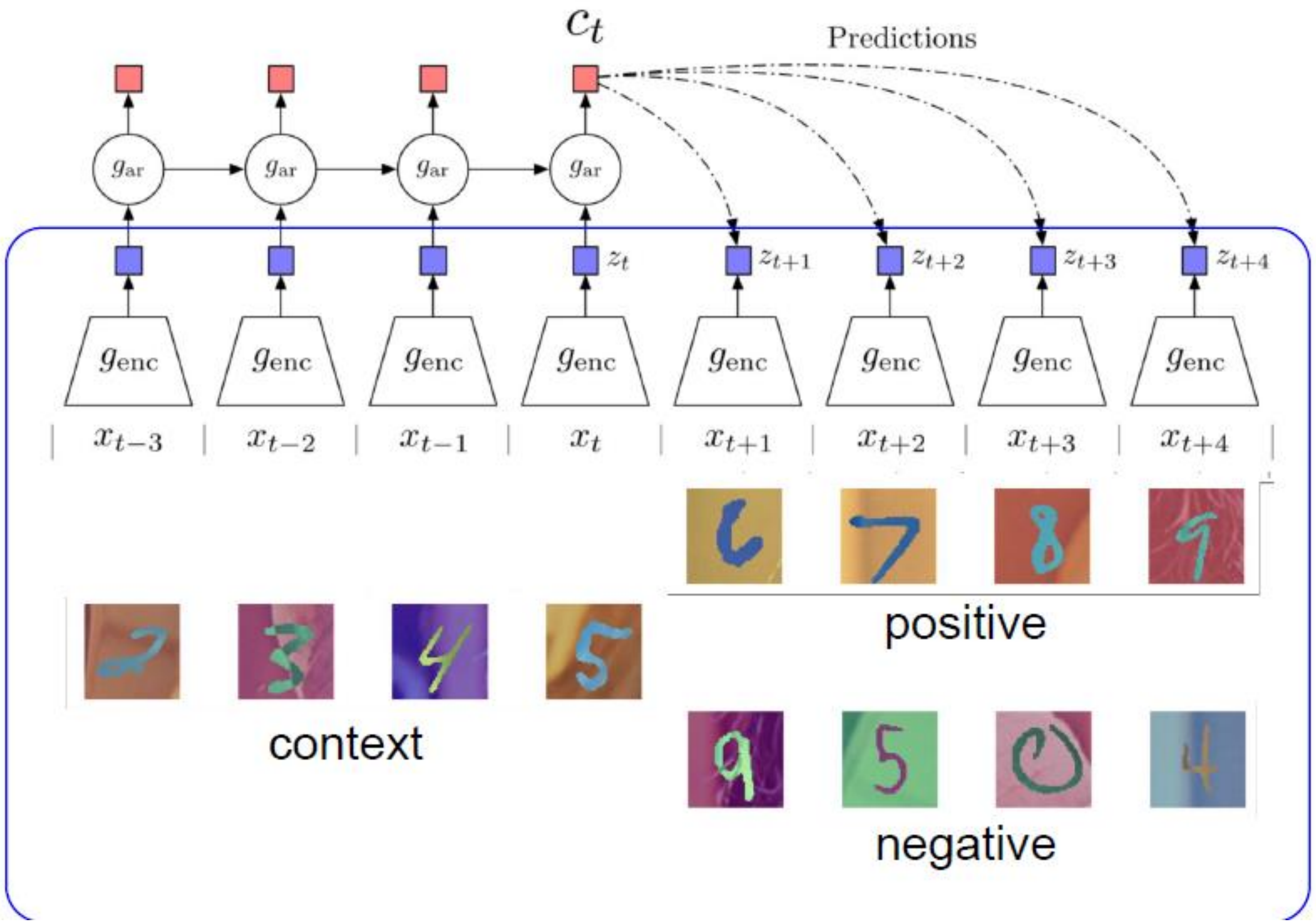


Figure 2: t-SNE visualization of audio (speech) representations for a subset of 10 speakers (out of 251). Every color represents a different speaker.

CPCs capture both speaker identity and speech contents.

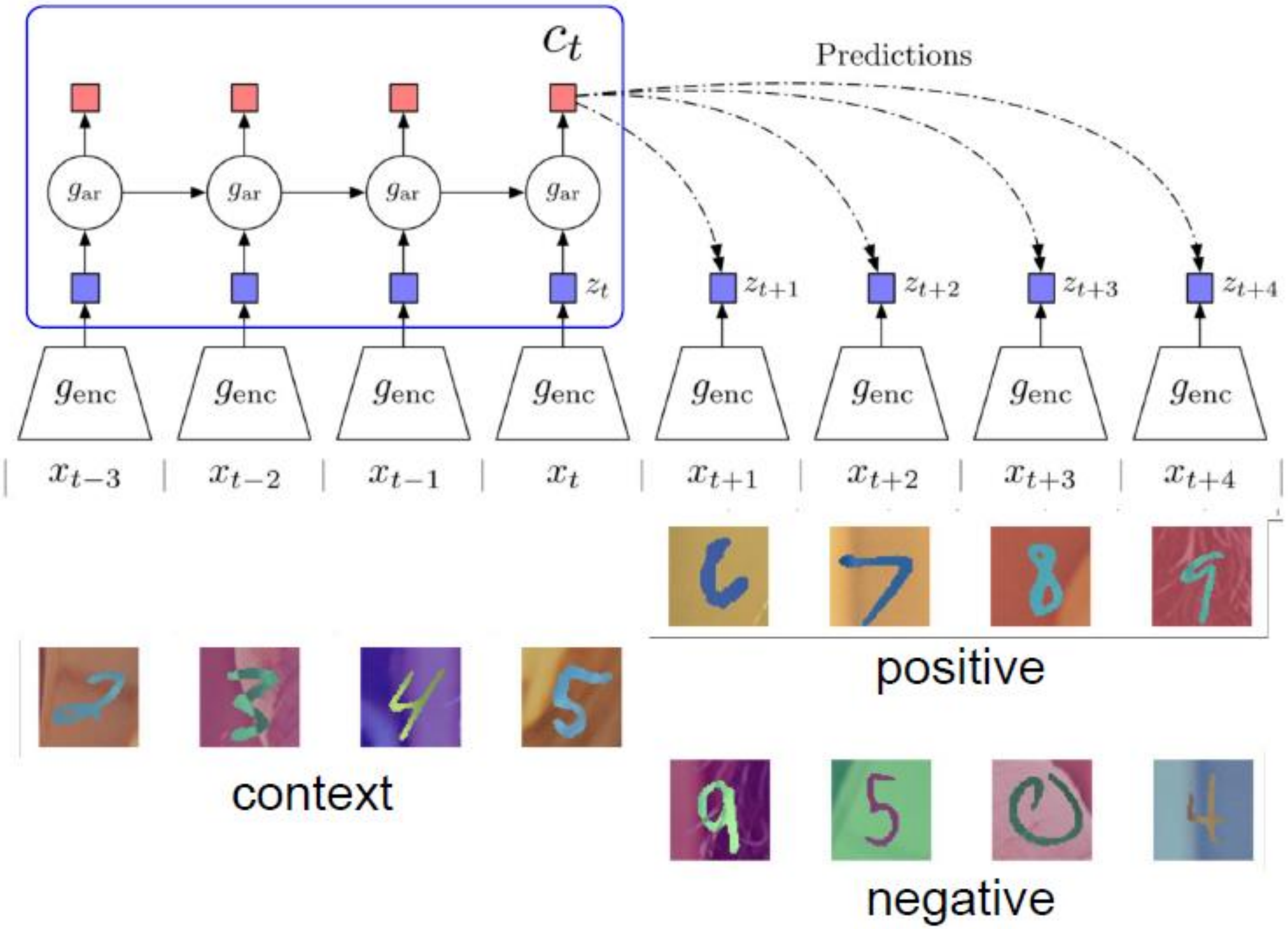
Thank you !!

Contrastive Predictive Coding (CPC)



1. Encode all samples in a sequence into vectors $\boldsymbol{z_t = g_{enc}(x_t)}$

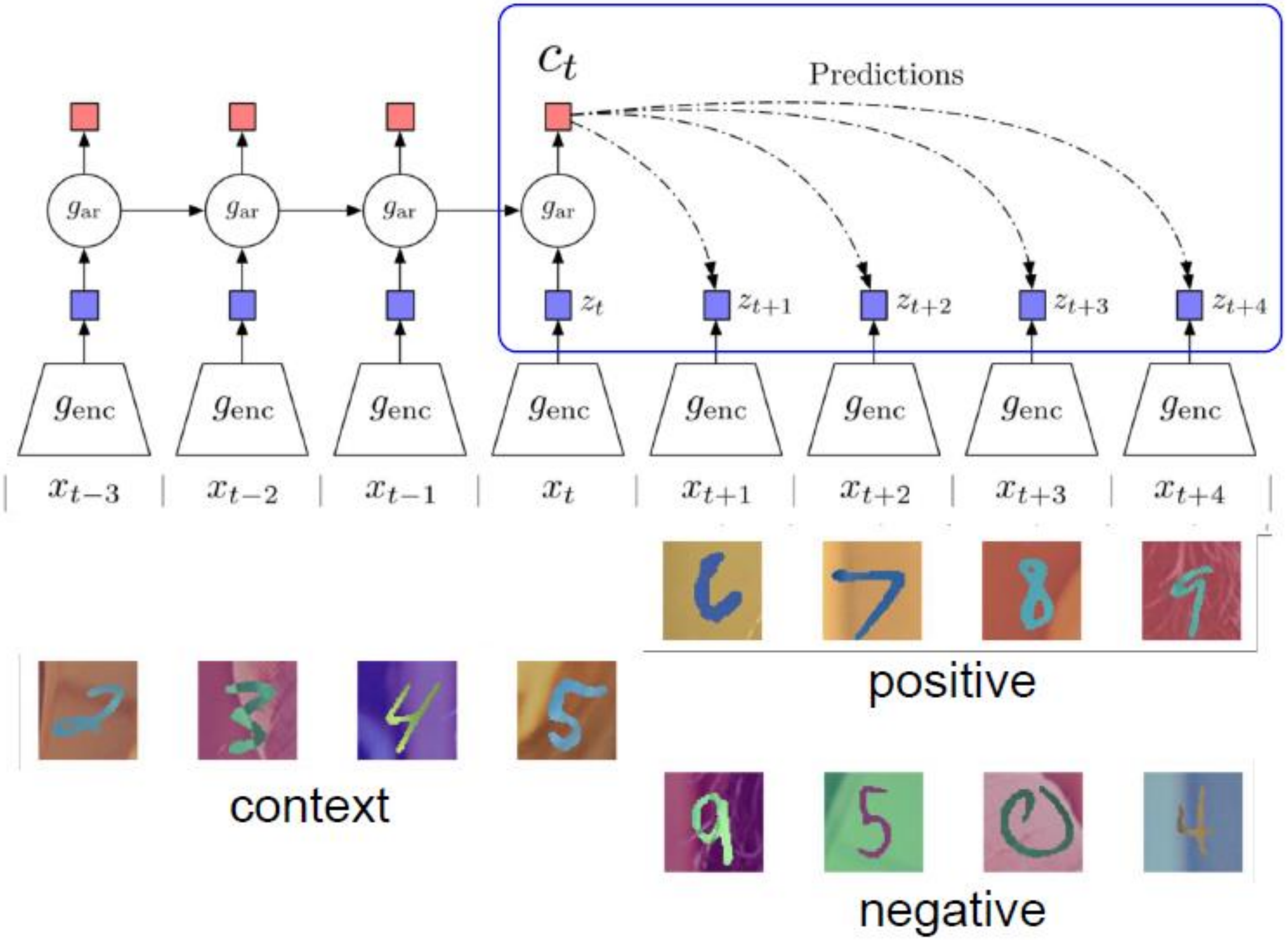
Contrastive Predictive Coding (CPC)



1. Encode all samples in a sequence into vectors $\boldsymbol{z_t = g_{enc}(x_t)}$

2. Summarize context (e.g., half of a sequence) into a context code $\boldsymbol{c_t}$ using an auto-regressive model ($\boldsymbol{g_{ar}}$). The original paper uses GRU-RNN here.

Contrastive Predictive Coding (CPC)



1. Encode all samples in a sequence into vectors $\boldsymbol{z_t} = \boldsymbol{g_{enc}(x_t)}$

2. Summarize context (e.g., half of a sequence) into a context code $\boldsymbol{c_t}$ using an auto-regressive model ($\boldsymbol{g_{ar}}$)

3. Compute InfoNCE loss between the context $\boldsymbol{c_t}$ and future code $\boldsymbol{z_{t+k}}$ using the following time-dependent score function:

$$s_k(z_{t+k}, c_t) = z_{t+k}^T W_k c_t$$

, where W_k is a trainable matrix.