# Representation Learning with Contrastive Predictive Coding

Aaron van den Oord, Yazhe Li, Oriol Vinyals
DeepMind

**Presented by: Desh Raj**

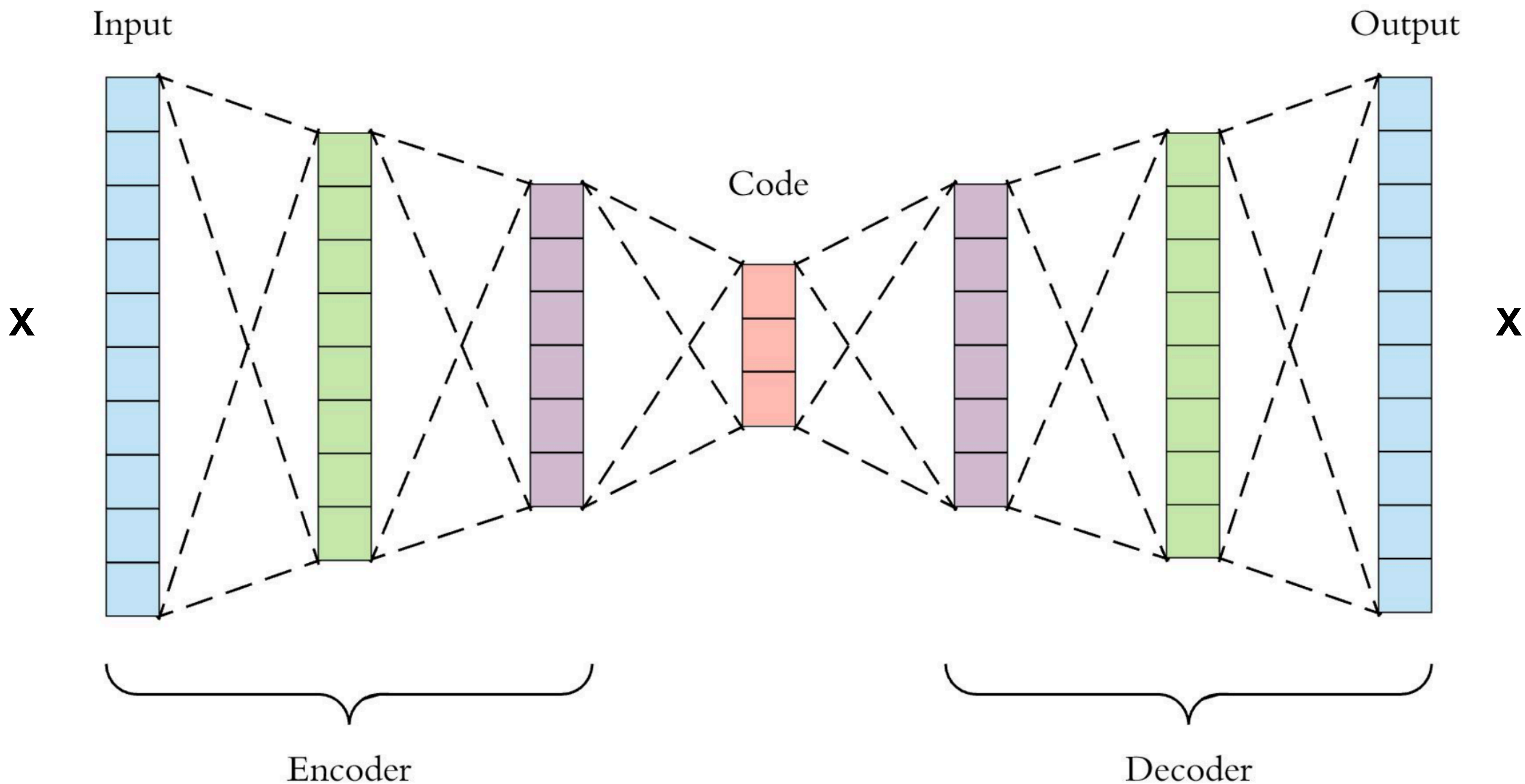
# Outline

- Why do we need general representations?
- The CPC model
- Understanding the loss function
- Experiments with speech data

# Why general representations?

- Manually specifying features is difficult and time-consuming.
- Features should be general—not specialized towards a single supervised task.
- E.g. features that are useful to transcribe human speech may be less suited for speaker identification.

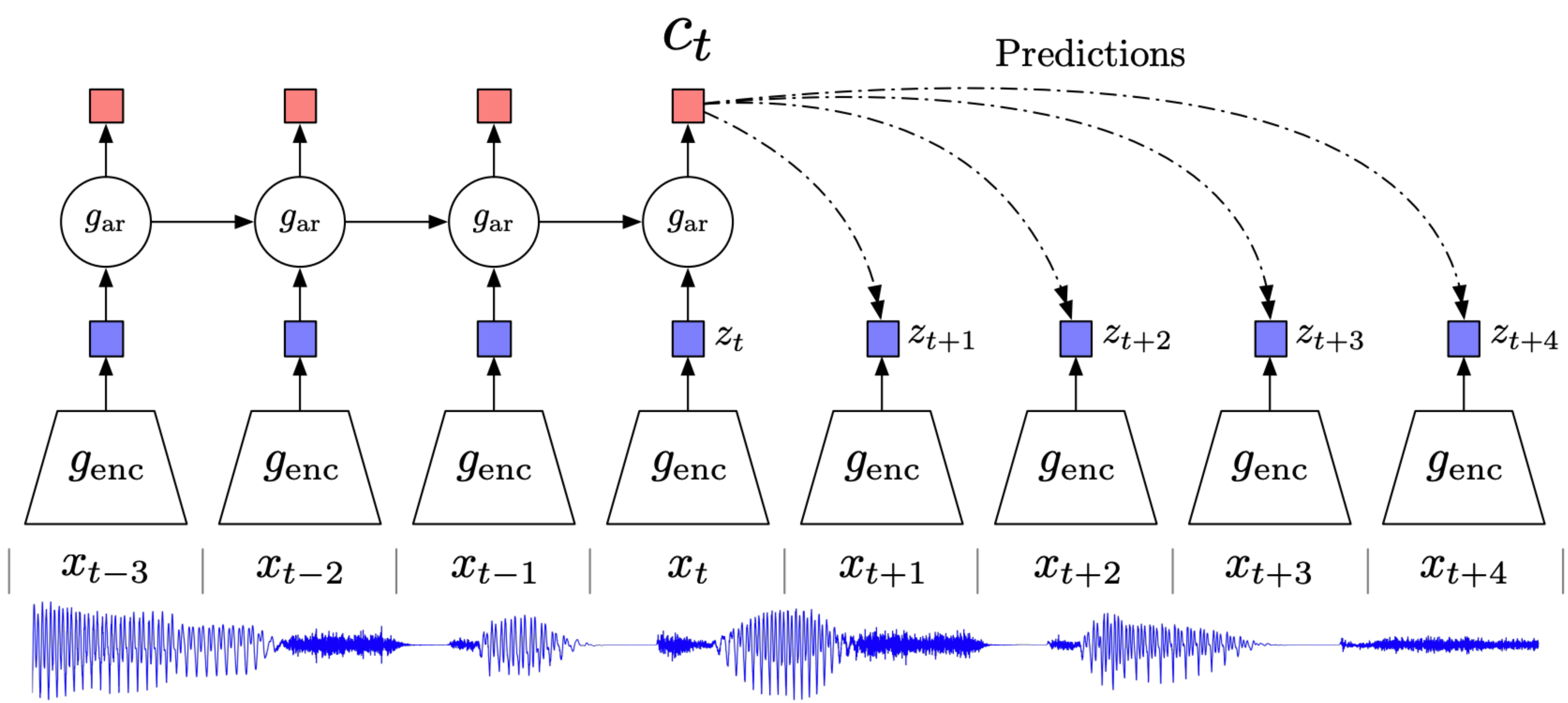
# Background: Autoencoders



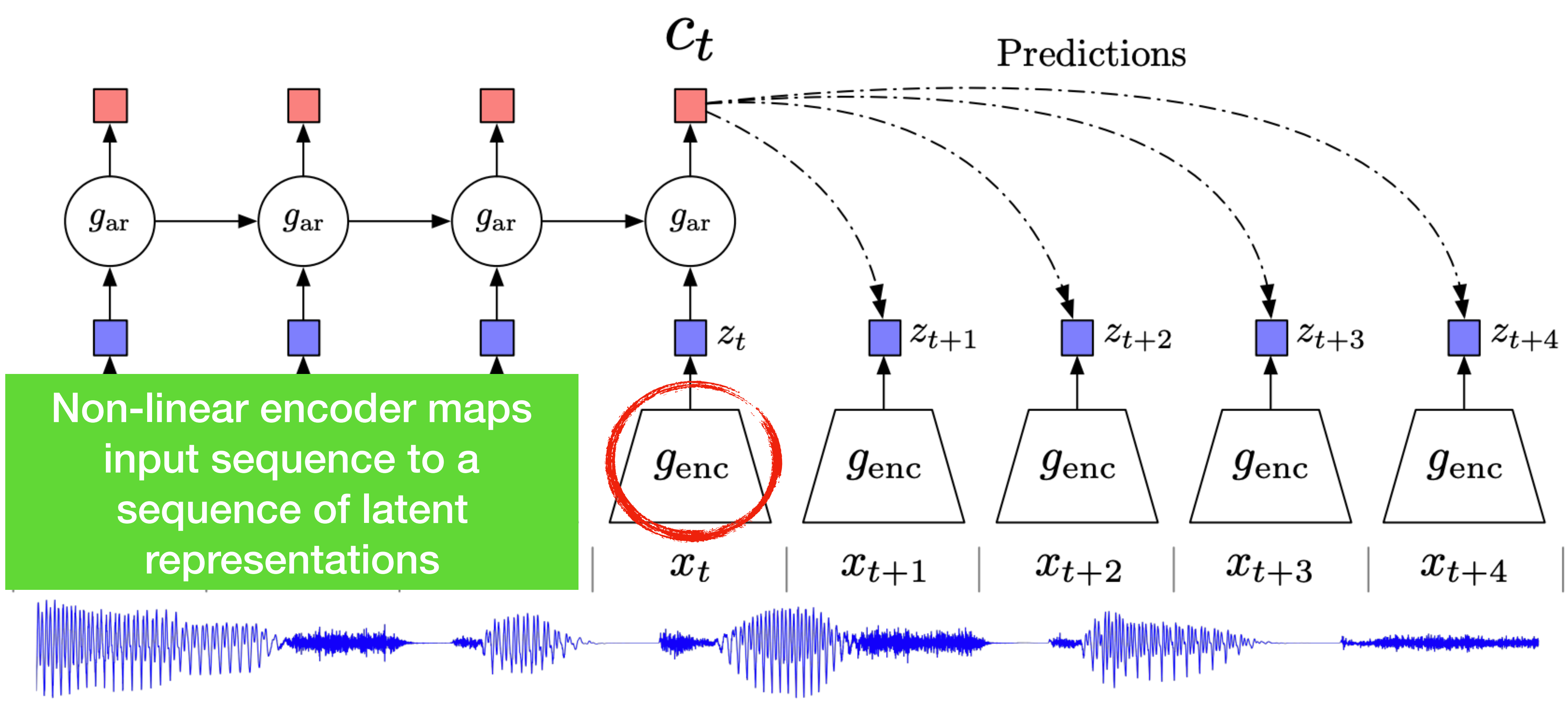
# Motivation and intuition

- Learn representations that encode underlying shared information between different parts of the signal.
- How to do this?
- Given a context, predict a future target.
- Sounds familiar? LM-based representation learning.

# The CPC model

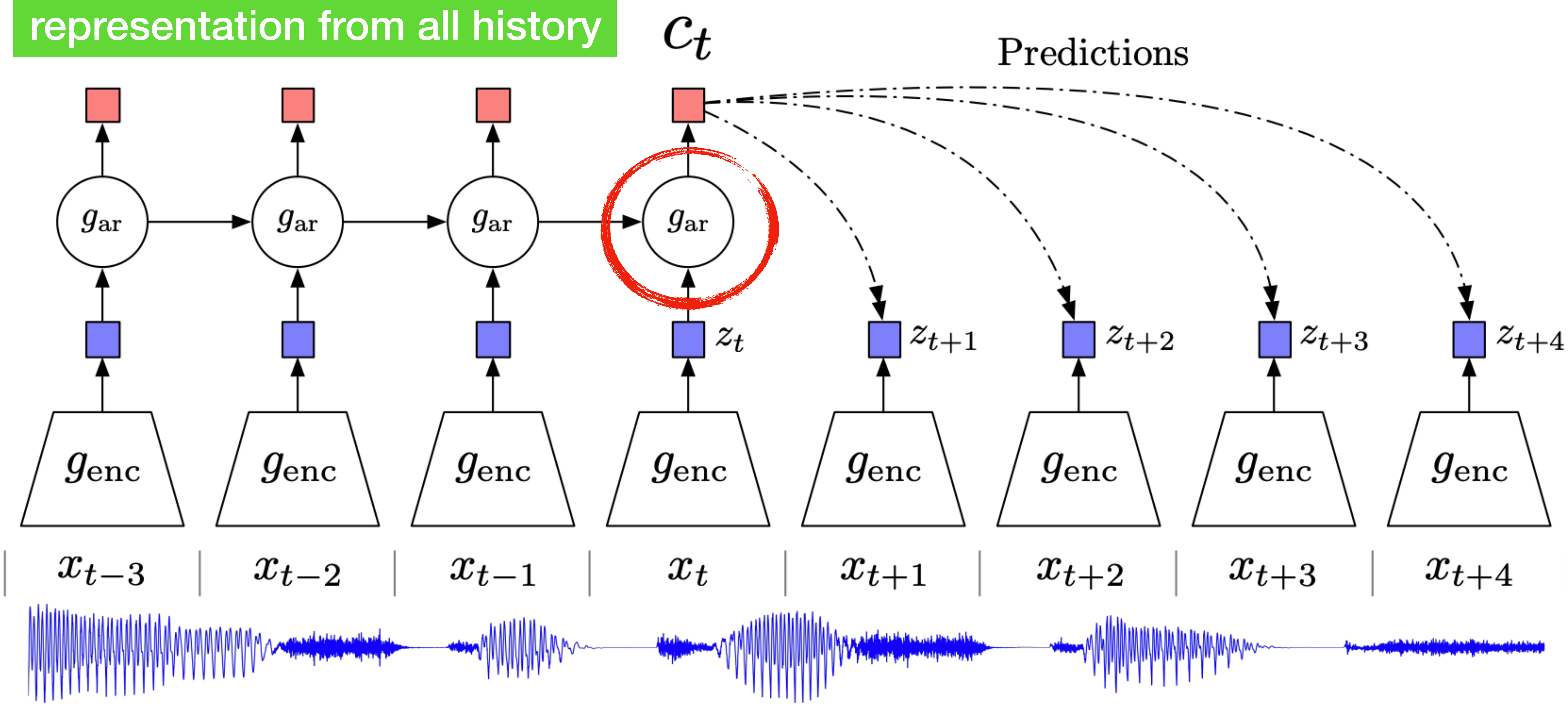


# The CPC model

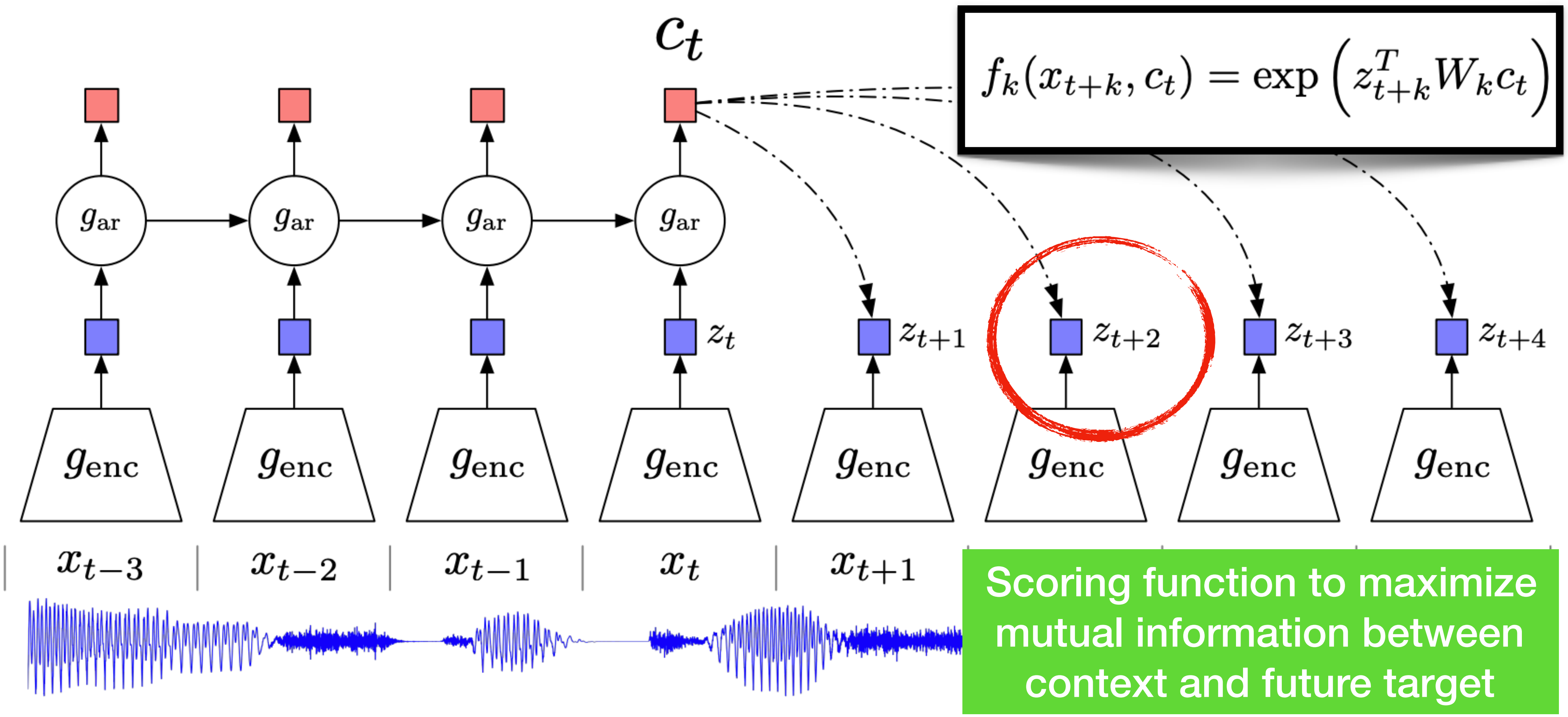


# The CPC model

Autoregressive model produces context representation from all history



# The CPC model



# The CPC model

- Any encoder and autoregressive model can be used in this framework.
- The authors used:
    - Strided convolutional layers with resnet blocks for encoder
    - GRUs for the autoregressive model

# The loss function

- Instead of directly predicting a target given a context, we try to maximize the **mutual information** between them.

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

# The loss function

- **InfoNCE loss**

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

**This is optimized when:**

$$f_k(x_{t+k}, c_t) \propto \frac{p(x_{t+k}|c_t)}{p(x_{t+k})}$$

- How does this loss maximize the mutual information?

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

- How does this loss maximize the mutual information?

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

$$\mathcal{L}_{\mathrm{N}}^{\mathrm{opt}} = -\mathop{\mathbb{E}}_{X} \log \left[\frac{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})}}{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})} + \sum_{x_j \in X_{\mathrm{neg}}} \frac{p(x_j|c_t)}{p(x_j)}}\right]$$

**Use proportionality condition**

- How does this loss maximize the mutual information?

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

$$\mathcal{L}_{\mathrm{N}}^{\mathrm{opt}} = -\mathop{\mathbb{E}}_{X} \log \left[\frac{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})}}{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})} + \sum_{x_j \in X_{\mathrm{neg}}} \frac{p(x_j|c_t)}{p(x_j)}}\right]$$

$$= \mathop{\mathbb{E}}_{X} \log \left[1 + \frac{p(x_{t+k})}{p(x_{t+k}|c_t)} \sum_{x_j \in X_{\mathrm{neg}}} \frac{p(x_j|c_t)}{p(x_j)}\right]$$

Take -ve inside log

- How does this loss maximize the mutual information?

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

$$\begin{aligned}
\mathcal{L}_{\mathrm{N}}^{\mathrm{opt}} &= -\mathop{\mathbb{E}}_{X} \log \left[\frac{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})}}{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})} + \sum_{x_j \in X_{\mathrm{neg}}} \frac{p(x_j|c_t)}{p(x_j)}}\right] \\
&= \mathop{\mathbb{E}}_{X} \log \left[1 + \frac{p(x_{t+k})}{p(x_{t+k}|c_t)} \sum_{x_j \in X_{\mathrm{neg}}} \frac{p(x_j|c_t)}{p(x_j)}\right] \\
&\approx \mathop{\mathbb{E}}_{X} \log \left[1 + \frac{p(x_{t+k})}{p(x_{t+k}|c_t)} (N-1) \mathop{\mathbb{E}}_{x_j} \frac{p(x_j|c_t)}{p(x_j)}\right]
\end{aligned}$$

**This approximation becomes more accurate as N increases, so it is preferable to use large negative samples**

- How does this loss maximize the mutual information?

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

$$\begin{aligned}
\mathcal{L}_{\mathrm{N}}^{\mathrm{opt}} &= -\mathop{\mathbb{E}}_{X} \log \left[\frac{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})}}{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})} + \sum_{x_j \in X_{\mathrm{neg}}} \frac{p(x_j|c_t)}{p(x_j)}}\right] \\
&= \mathop{\mathbb{E}}_{X} \log \left[1 + \frac{p(x_{t+k})}{p(x_{t+k}|c_t)} \sum_{x_j \in X_{\mathrm{neg}}} \frac{p(x_j|c_t)}{p(x_j)}\right] \\
&\approx \mathop{\mathbb{E}}_{X} \log \left[1 + \frac{p(x_{t+k})}{p(x_{t+k}|c_t)} (N-1) \mathop{\mathbb{E}}_{x_j} \frac{p(x_j|c_t)}{p(x_j)}\right] \qquad = 1 \\
&= \mathop{\mathbb{E}}_{X} \log \left[1 + \frac{p(x_{t+k})}{p(x_{t+k}|c_t)} (N-1)\right]
\end{aligned}$$

- How does this loss maximize the mutual information?

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

$$\begin{aligned}
\mathcal{L}_{\mathrm{N}}^{\mathrm{opt}} &= -\mathop{\mathbb{E}}_{X} \log \left[ \frac{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})}}{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})} + \sum_{x_j \in X_{\mathrm{neg}}} \frac{p(x_j|c_t)}{p(x_j)}} \right] \\
&= \mathop{\mathbb{E}}_{X} \log \left[ 1 + \frac{p(x_{t+k})}{p(x_{t+k}|c_t)} \sum_{x_j \in X_{\mathrm{neg}}} \frac{p(x_j|c_t)}{p(x_j)} \right] \\
&\approx \mathop{\mathbb{E}}_{X} \log \left[ 1 + \frac{p(x_{t+k})}{p(x_{t+k}|c_t)} (N-1) \mathop{\mathbb{E}}_{x_j} \frac{p(x_j|c_t)}{p(x_j)} \right] \\
&= \mathop{\mathbb{E}}_{X} \log \left[ 1 + \frac{p(x_{t+k})}{p(x_{t+k}|c_t)} (N-1) \right] \\
&\geq \mathop{\mathbb{E}}_{X} \log \left[ \frac{p(x_{t+k})}{p(x_{t+k}|c_t)} N \right] \\
&= -I(x_{t+k}, c_t) + \log(N),
\end{aligned}$$

- How does this loss maximize the mutual information?

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

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

- How does this loss maximize the mutual information?

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

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

**Minimizing the loss function maximizes the lower bound on mutual information**

- How does this loss maximize the mutual information?

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

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

**Higher value of N is also useful here to increase mutual information**

# Experimental results

- Experiments with speech, text, image, and reinforcement learning.

- Here, I'll only present results on **speech data**.

- Corpus: 100-hour subset of LibriSpeech.

# Experimental results

| 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.**

# Experimental results

| 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.

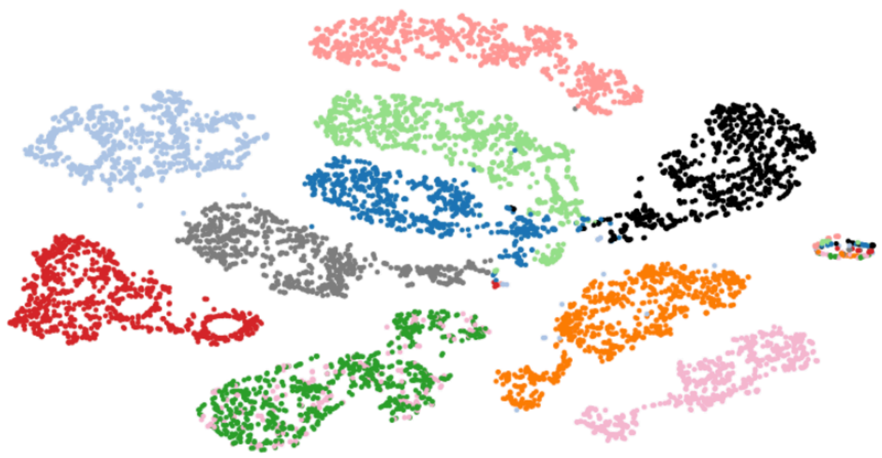


Figure 2: t-SNE visualization of audio (speech) representations for a subset of 10 speakers (out of 251). Every color represents a different speaker.

**CPCs capture both speaker identity and speech contents.**

# Key takeaways

- Autoencoding through **prediction**, not reconstruction.
- Capture **high level content**, ignore noise.
- Use **sequences** to “mimic” labeled data.

```
sys.exit(0)
```