# Representation Learning with Contrastive Predictive Coding

Aaron van den Oord - Yazhe Li - Oriol Vinyals
Presenter: Thang Duong

# Outline

1. Preliminary: Contrastive Learning and Predictive Coding

2. Representation learning motivation

   - Conditional on the context (i.e. Supervised learning)
   - Unconditional (i.e. Un/Self-supervised learning)

3. Problem setup and assumptions

4. Contrastive Predictive Coding and InfoNCE

5. Experiments

6. Discussion

# 1. Preliminary: Contrastive Learning and Predictive Coding

- Contrastive + Predictive Coding
- Contrastive Learning: from triplet loss to Noise Contrastive Estimation (NCE). Lil’log refs: [1] [2]

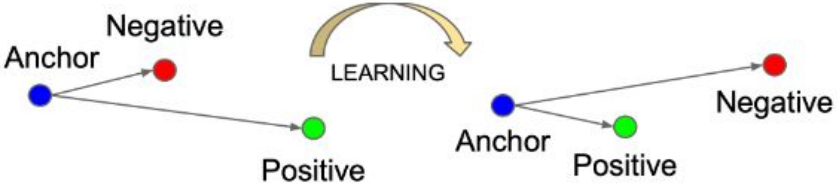
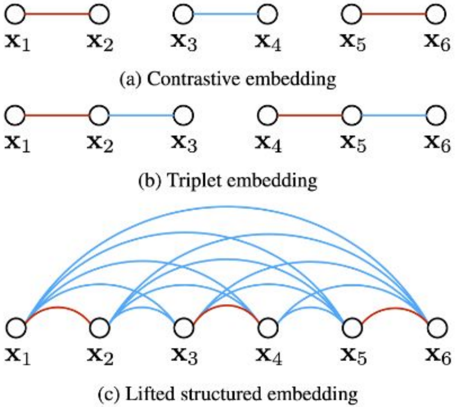


Fig. 1. Illustration of triplet loss given one positive and one negative per anchor. (Image source: Schroff et al. 2015)



(a) Contrastive embedding

(b) Triplet embedding

(c) Lifted structured embedding

Fig. 2. Illustration compares contrastive loss, triplet loss and lifted structured loss. Red and blue edges connect similar and dissimilar sample pairs respectively. (Image source: Song et al. 2015)

# 1. Preliminary: Contrastive Learning and Predictive Coding

- Contrastive Learning:
  - NCE: construct a noise distribution over the negative samples (not just uniformly select negative samples)
  - Key factors (from empirical results):
    - Data augmentation for positive samples. SimCLR: random crop + random color distortion
    - Large batch size (diverse negative samples)
    - Hard Negative Mining (supervised?)

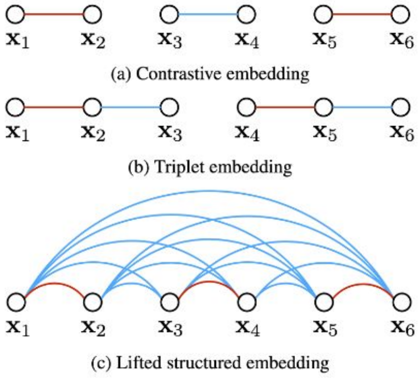


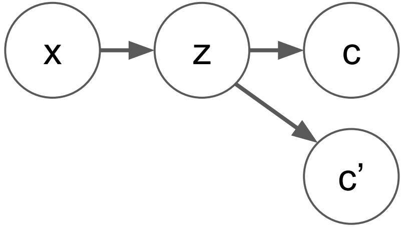
Fig. 2. Illustration compares contrastive loss, triplet loss and lifted structured loss. Red and blue edges connect similar and dissimilar sample pairs respectively. (Image source: Song et al. 2015)

# 1. Preliminary: Contrastive Learning and Predictive Coding

- Predictive Coding
  - Theory of brain function (wiki): build an internal model to predict the input signals and update it by compare with the true signals.
  - Back prop is a special case
  - Used in sequential data tasks: speech, video, RL, etc.

# 2. Representation learning motivation

- Motivation: transfer representation to reduce the sample complexity



$$\min_{\theta\in\mathbb{R}^d} L = \sum_i < x_i, \theta >,\ x_i \in \mathbb{R}^d$$

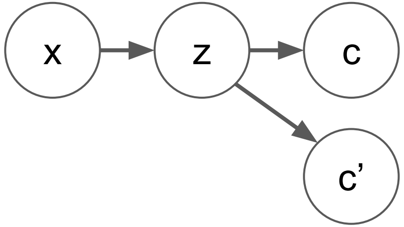
$$<=> \min_{B\in\mathbb{R}^{d\times m}, w\in\mathbb{R}^m} = \sum_i < x_i, Bw >$$

$$= \sum_i < (x_i^\top B)^\top, w >$$

# 2. Representation learning motivation

- Conditional on the context (i.e. Supervised learning):
  - Layers in NNs trained with labeled data: future, missing, or contextual information
  - p(x|c): thousands bits of info in an images but only 10 bits in the labels (1024 classes)

- Unconditional (i.e. Un/Self-supervised learning):
  - E.g. predicting missing words, colorization from grayscale
  - Generative losses are better compared to unimodal losses (MSE, Cross-entropy) in learning (representation) for high-dimensional data



- Objective:
  - Maximize the mutual information I(z; c)
  - Different from the Information bottleneck: I(z; c) - I(x; z)

# 3. Problem setup and assumptions

- Architecture: encoder + an autoregressive model for summarizing previous latent info (Non-Markovian)

JEPA: Markovian and $c_t = z_{t+1}$

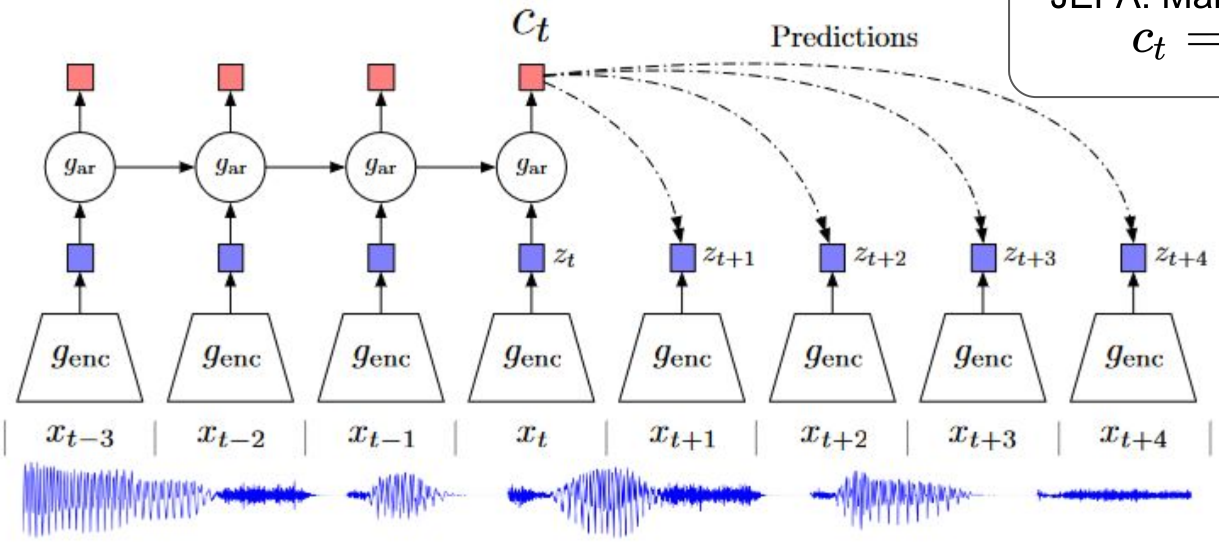
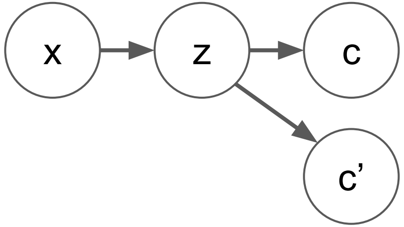


Figure 1: Overview of Contrastive Predictive Coding, the proposed representation learning approach. Although this figure shows audio as input, we use the same setup for images, text and reinforcement learning.

# 4. Contrastive Predictive Coding and InfoNCE



- Objective:
  - Maximize the mutual information I(z; c)
  - Different from the Information bottleneck: I(z; c) - I(x; z)

- Actual objective: maximize a function f that’s proportional to the density ratio:

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

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

k: future step

Why log-bilinear model?

- Consider x=z and W=I. Then, f is maximized when z = c (Cauchy-Schwarz inequality)

# 4. Contrastive Predictive Coding and InfoNCE

X: (N-1) negative samples and one positive sample

## InfoNCE Loss and Mutual Information Estimation

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

Assume data from each step comes from a distribution?

- The noise/proposal distribution for negative samples: $p\left(x_{t+k}\right)$
- Optimizing this loss as a categorical cross-entropy to classify pos/neg samples:

$$\begin{aligned} p\left(d=i \mid X, c_t\right) &= \frac{p\left(x_i \mid c_t\right) \prod_{l \neq i} p\left(x_l\right)}{\sum_{j=1}^{N} p\left(x_j \mid c_t\right) \prod_{l \neq j} p\left(x_l\right)} \\ &= \frac{\frac{p(x_i \mid c_t)}{p(x_i)}}{\sum_{j=1}^{N} \frac{p(x_j \mid c_t)}{p(x_j)}}. \end{aligned}$$

=> $f_k\left(x_{t+k}, c_t\right) \propto \frac{p(x_{t+k} \mid c_t)}{p(x_{t+k})}$

# 4. Contrastive Predictive Coding and InfoNCE

- Minimizing the InfoNCE loss maximizes a lower bound on mutual information

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

$$\begin{aligned}
\mathcal{L}_{\mathrm{N}}^{\mathrm{opt}} &= -\mathop{\mathbb{E}}_{X} \log \left[ \frac{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})}}{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})} + \sum_{x_j \in X_{\mathrm{neg}}} \frac{p(x_j|c_t)}{p(x_j)}} \right] && (6) \\
&= \mathop{\mathbb{E}}_{X} \log \left[ 1 + \frac{p(x_{t+k})}{p(x_{t+k}|c_t)} \sum_{x_j \in X_{\mathrm{neg}}} \frac{p(x_j|c_t)}{p(x_j)} \right] && (7) \\
&\approx \mathop{\mathbb{E}}_{X} \log \left[ 1 + \frac{p(x_{t+k})}{p(x_{t+k}|c_t)} (N-1) \mathop{\mathbb{E}}_{x_j} \frac{p(x_j|c_t)}{p(x_j)} \right] && (8) \\
&= \mathop{\mathbb{E}}_{X} \log \left[ 1 + \frac{p(x_{t+k})}{p(x_{t+k}|c_t)} (N-1) \right] && (9) \\
&\geq \mathop{\mathbb{E}}_{X} \log \left[ \frac{p(x_{t+k})}{p(x_{t+k}|c_t)} N \right] && (10) \\
&= -I(x_{t+k}, c_t) + \log(N), && (11)
\end{aligned}$$

# 4. Contrastive Predictive Coding and InfoNCE

- InfoNCE is equivalent to the MINE estimator (up to a constant)

$$\begin{aligned}
\mathop{\mathbb{E}}_{X}\left[\log \frac{f(x,c)}{\sum_{x_j \in X} f(x_j,c)}\right] &= \mathop{\mathbb{E}}_{(x,c)}\left[F(x,c)\right] - \mathop{\mathbb{E}}_{(x,c)}\left[\log \sum_{x_j \in X} e^{F(x_j,c)}\right] \qquad (12) \\
&= \mathop{\mathbb{E}}_{(x,c)}\left[F(x,c)\right] - \mathop{\mathbb{E}}_{(x,c)}\left[\log \left(e^{F(x,c)} + \sum_{x_j \in X_{\text{neg}}} e^{F(x_j,c)}\right)\right] \qquad (13) \\
&\leq \mathop{\mathbb{E}}_{(x,c)}\left[F(x,c)\right] - \mathop{\mathbb{E}}_{c}\left[\log \sum_{x_j \in X_{\text{neg}}} e^{F(x_j,c)}\right] \qquad (14) \\
&= \mathop{\mathbb{E}}_{(x,c)}\left[F(x,c)\right] - \mathop{\mathbb{E}}_{c}\left[\log \frac{1}{N-1} \sum_{x_j \in X_{\text{neg}}} e^{F(x_j,c)} + \log(N-1)\right] \\
&\qquad (15)
\end{aligned}$$

- “Using MINE directly gave identical performance when the task was nontrivial, but became very unstable if the target was easy to predict from the context (e.g., when predicting a single step in the future and the target overlaps with the context).”

$$c_t = z_{t+1}$$

# 5. Experiments

Can use both z and c as representation vectors for downstream tasks

GRUs for the autoregressive model

resnet for the encoder

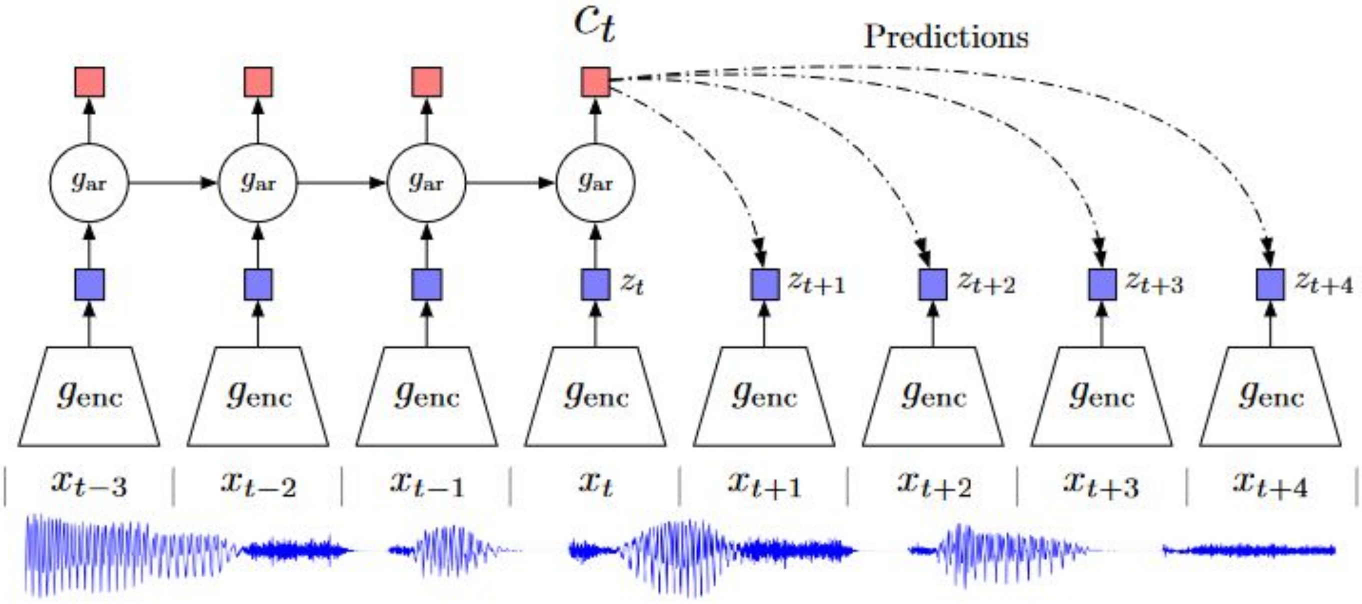


Figure 1: Overview of Contrastive Predictive Coding, the proposed representation learning approach. Although this figure shows audio as input, we use the same setup for images, text and reinforcement learning.

# 5. Experiments

- Audio: 100h of LibriSpeech. Generate label by force-aligned phone sequences with Kaldi toolkit
- 2 tasks: Phone classification and Speaker classification (linear last layer)

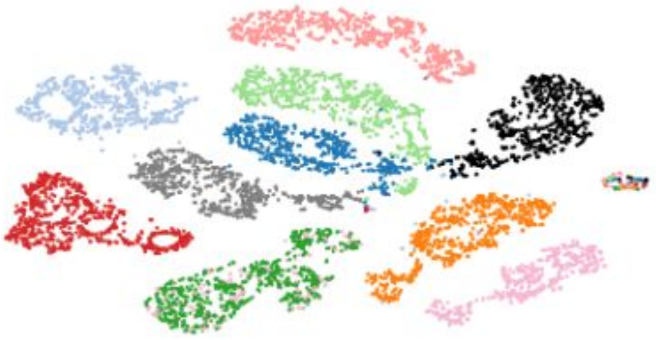


Figure 2: t-SNE visualization of audio (speech) representations for a subset of 10 speakers (out of 251). Every color represents a different speaker.

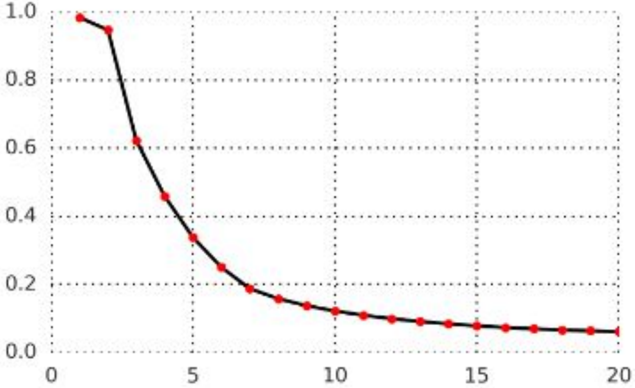
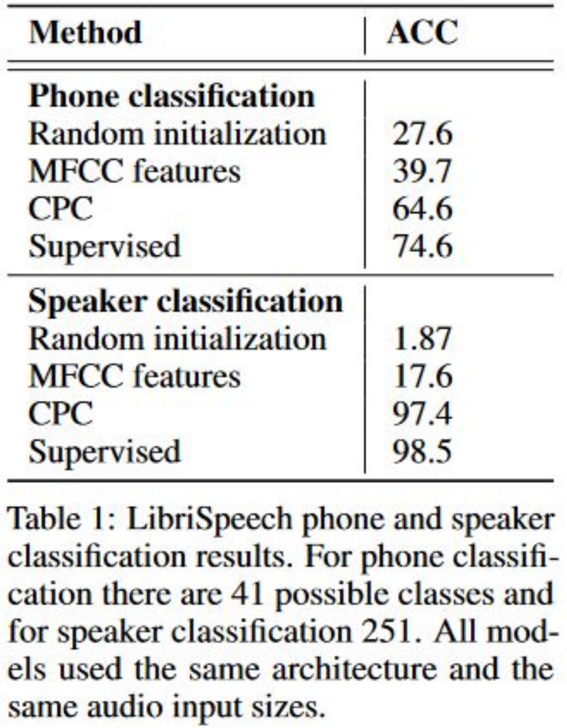


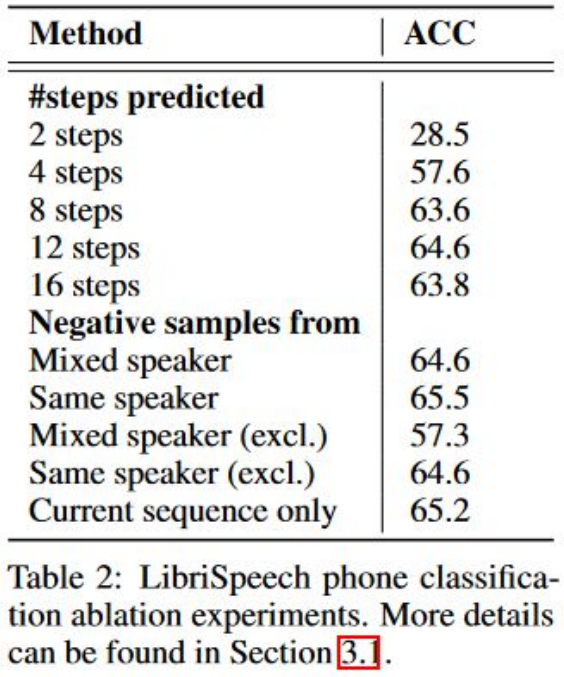
Figure 3: Average accuracy of predicting the positive sample in the contrastive loss for 1 to 20 latent steps in the future of a speech waveform. The model predicts up to 200ms in the future as every step consists of 10ms of audio.

# 5. Experiments



| 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.



| Method | ACC |
|---|---|
| **#steps predicted** | |
| 2 steps | 28.5 |
| 4 steps | 57.6 |
| 8 steps | 63.6 |
| 12 steps | 64.6 |
| 16 steps | 63.8 |
| **Negative samples from** | |
| Mixed speaker | 64.6 |
| Same speaker | 65.5 |
| Mixed speaker (excl.) | 57.3 |
| Same speaker (excl.) | 64.6 |
| Current sequence only | 65.2 |

Table 2: LibriSpeech phone classification ablation experiments. More details can be found in Section 3.1.

A bit counter intuitive

- “Table 2 … showing that predicting multiple steps is important for learning useful features”

# 5. Experiments

Vision:

- ImageNet data with augmentation (crop + flip).
- ResNet v2 101 architecture for encoder (not pretrained)
- Linear layer on top after unsupervised learning
- PixelCNN autoregressive model.
- Task: crop the (grayscale converted) image into overlapping patches. Use these patches as the sequential data and try to predict the activations of the future patches.

# 5. Experiments

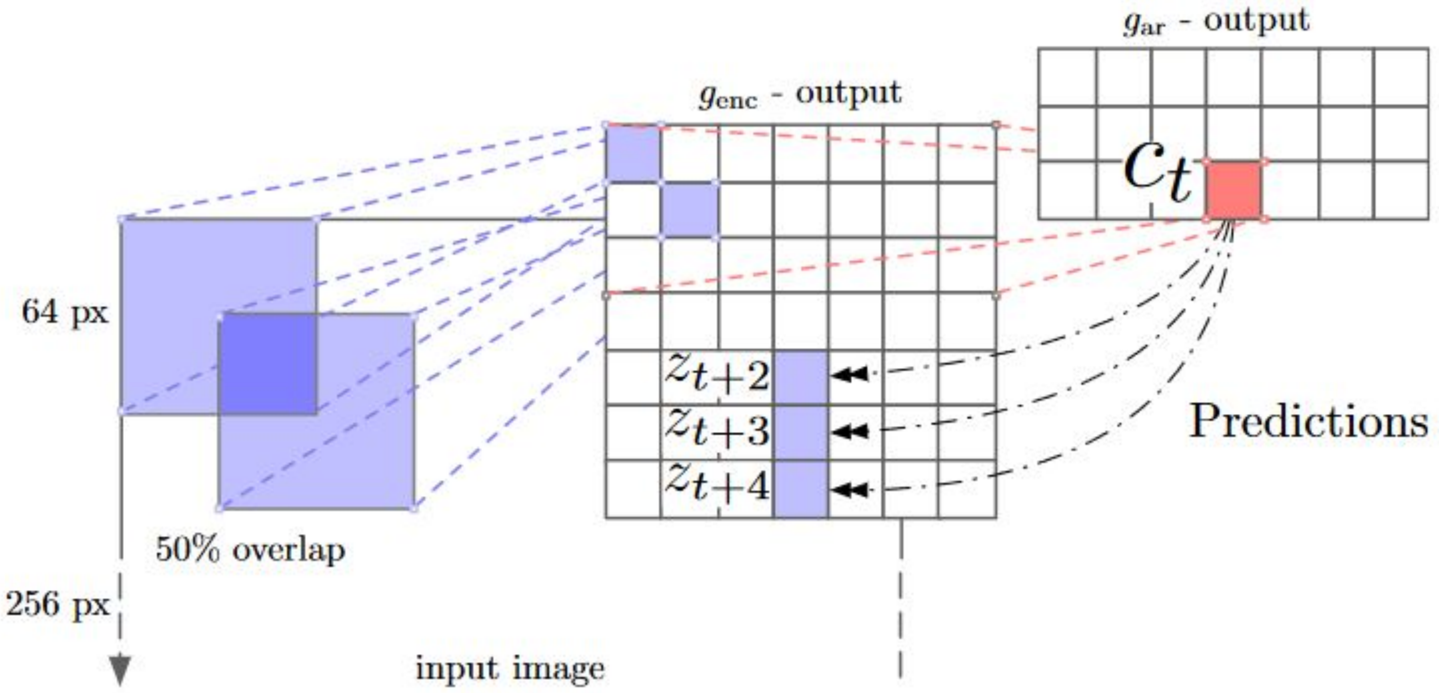


Figure 4: Visualization of Contrastive Predictive Coding for images (2D adaptation of Figure 1).

# 5. Experiments

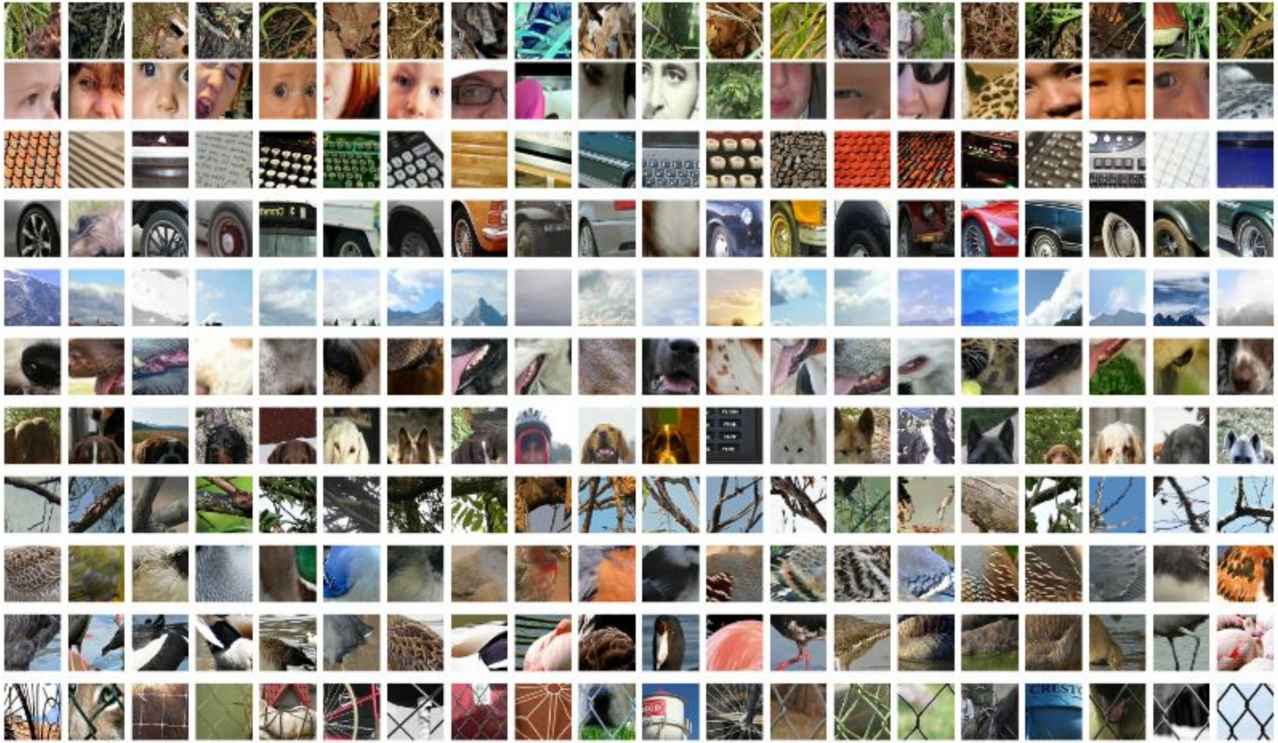


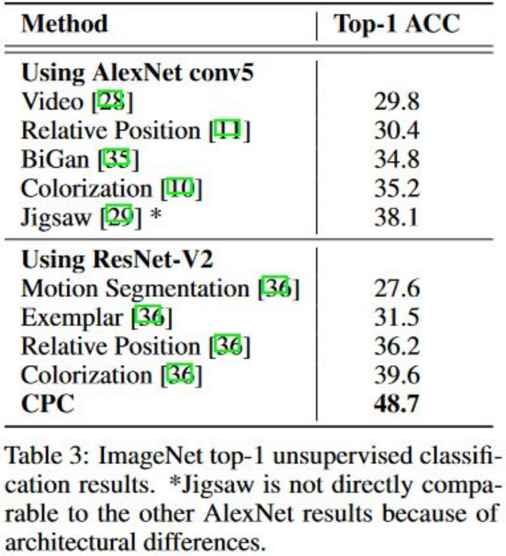
Figure 5: Every row shows image patches that activate a certain neuron in the CPC architecture.

# 5. Experiments

Natural Language:

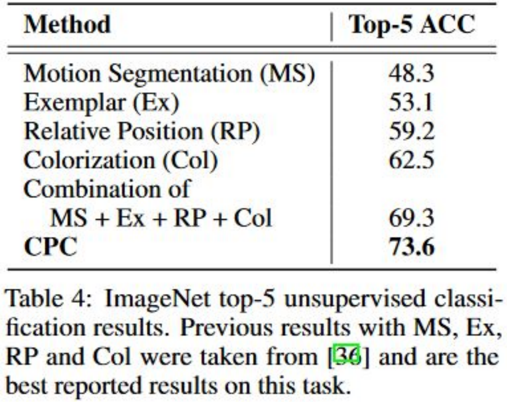
- BookCorpus dataset
- Using CPC representations for a set of classification tasks
  - Movie review sentiment (MR)
  - Customer product reviews (CR)
  - Subjectivity/objectivity [45]
  - Opinion polarity (MPQA)
  - Question-type classification (TREC)
- Logistic regression classifier on top
- Simple 1D CNN for encoder + GRU for autoregressive model

# 5. Experiments



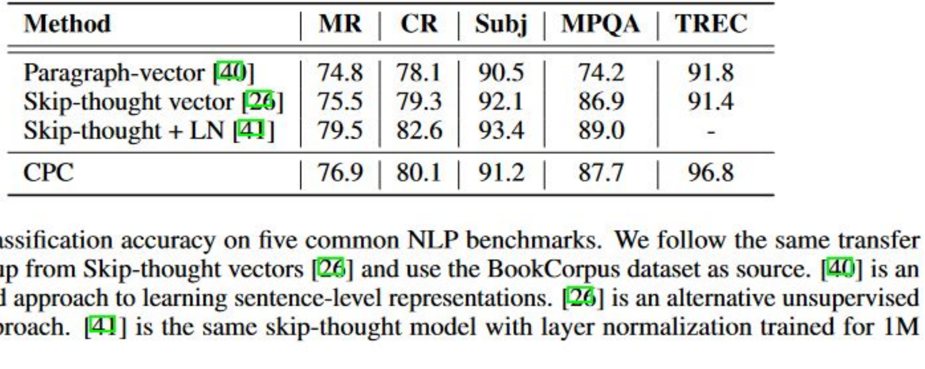
| Method | Top-1 ACC |
|---|---|
| **Using AlexNet conv5** | |
| Video [28] | 29.8 |
| Relative Position [11] | 30.4 |
| BiGan [35] | 34.8 |
| Colorization [10] | 35.2 |
| Jigsaw [29] * | 38.1 |
| **Using ResNet-V2** | |
| Motion Segmentation [36] | 27.6 |
| Exemplar [36] | 31.5 |
| Relative Position [36] | 36.2 |
| Colorization [36] | 39.6 |
| **CPC** | **48.7** |

Table 3: ImageNet top-1 unsupervised classification results. *Jigsaw is not directly comparable to the other AlexNet results because of architectural differences.



| Method | Top-5 ACC |
|---|---|
| Motion Segmentation (MS) | 48.3 |
| Exemplar (Ex) | 53.1 |
| Relative Position (RP) | 59.2 |
| Colorization (Col) | 62.5 |
| Combination of<br>MS + Ex + RP + Col | 69.3 |
| **CPC** | **73.6** |

Table 4: ImageNet top-5 unsupervised classification results. Previous results with MS, Ex, RP and Col were taken from [36] and are the best reported results on this task.



| Method | MR | CR | Subj | MPQA | TREC |
|---|---|---|---|---|---|
| Paragraph-vector [40] | 74.8 | 78.1 | 90.5 | 74.2 | 91.8 |
| Skip-thought vector [26] | 75.5 | 79.3 | 92.1 | 86.9 | 91.4 |
| Skip-thought + LN [41] | 79.5 | 82.6 | 93.4 | 89.0 | - |
| CPC | 76.9 | 80.1 | 91.2 | 87.7 | 96.8 |

Table 5: Classification accuracy on five common NLP benchmarks. We follow the same transfer learning setup from Skip-thought vectors [26] and use the BookCorpus dataset as source. [40] is an unsupervised approach to learning sentence-level representations. [26] is an alternative unsupervised learning approach. [41] is the same skip-thought model with layer normalization trained for 1M iterations.

# 5. Experiments

## Reinforcement Learning

- 5 DeepMind Lab tasks
- Batched A2C agent as base model and add CPC as an auxiliary loss
  - Violating the Markov property? Same as the trick of using multiple frames as input in DQN paper
- No replay buffer (harder, requires good representation)
- CNN encoder + LSTM
- “Use the same encoder as in the baseline agent and only add the linear prediction mappings for the contrastive loss”
- “We do not use a replay buffer, so the predictions have to adapt to the changing behavior of the policy. The learned representation encodes a distribution over its future observations.”

# 5. Experiments

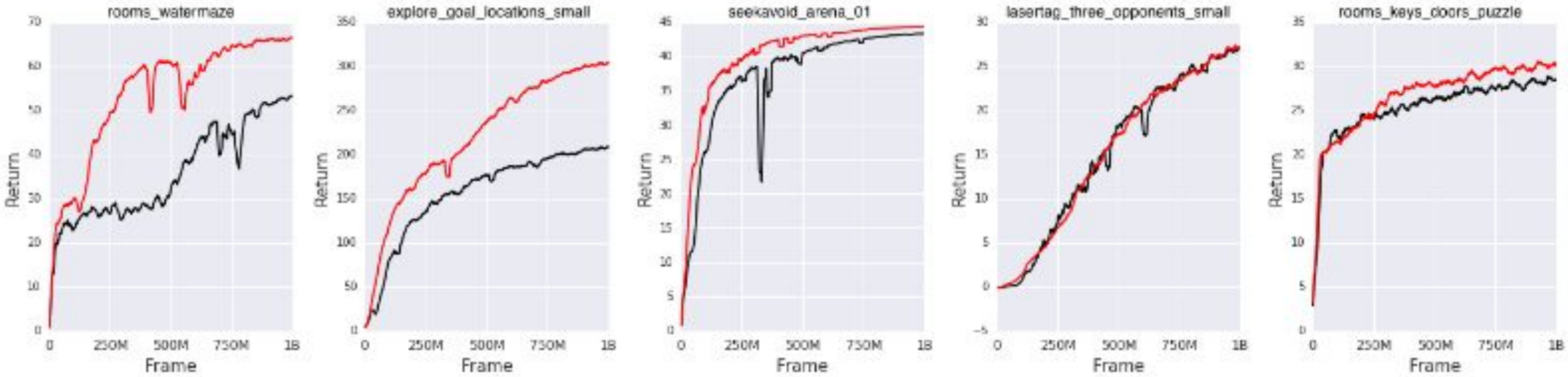


Figure 6: Reinforcement Learning results for 5 DeepMind Lab tasks used in [50]. Black: batched A2C baseline, Red: with auxiliary contrastive loss.

# 6. Discussion

- Not easy to learn a “good” autoregressive model here.
  - 1-2 directional LSTM vs Transformer. The authors suggested alternative: masked convnet and self-attention nets.
- Each word has less information compared to an image => may be easier to model a complex graph this way.
- For high-dimensional data (sounds, image, etc.) => may work if the graph is simple (such as having Markov property + simple dynamic (e.g. low-rank, simple local dynamic))
- For RL: select action from the representation directly?
  - The architecture roughly describes using RNN to solve RL (by MLE with backprop through time). Even for simple toy examples, the performance is not very good.
  - With Markov property (i.e. JEPA) this turned into RL with rich observation (or Block MDP in particular)

# Summary

- Contrastive Predictive Coding (CPC): “combines autoregressive modeling and noise-contrastive estimation with intuitions from predictive coding to learn abstract representations in an unsupervised fashion”
- Simple + low computational requirement returns strong or SOTA results in a wide variety of domains: audio, images, natural language and reinforcement learning
- CPC extends the Noise Contrastive Estimation to InfoNCE, which is equivalent to Mutual Information lower bound, MINE (up to a constant). It has some connection with Information Bottleneck objective.