Putting An End to End-to-End: Gradient-Isolated Learning of Representations

Sindy Löwe, Peter O’Connor, Bastiaan S. Veeling
AMLab, University of Amsterdam

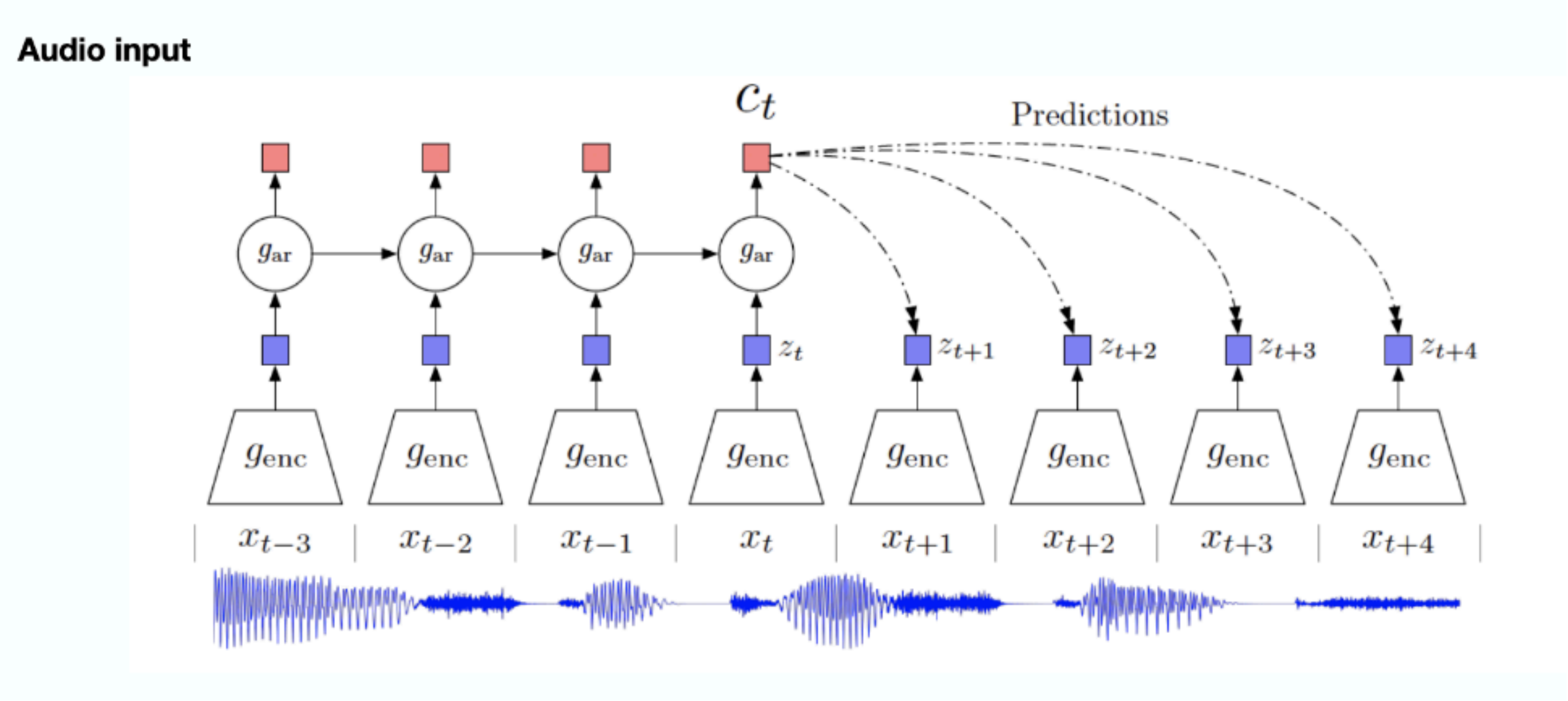
Honorable Mention Outstanding New Directions Paper Award, NeurIPS 2019

Presented by Hongming Shan

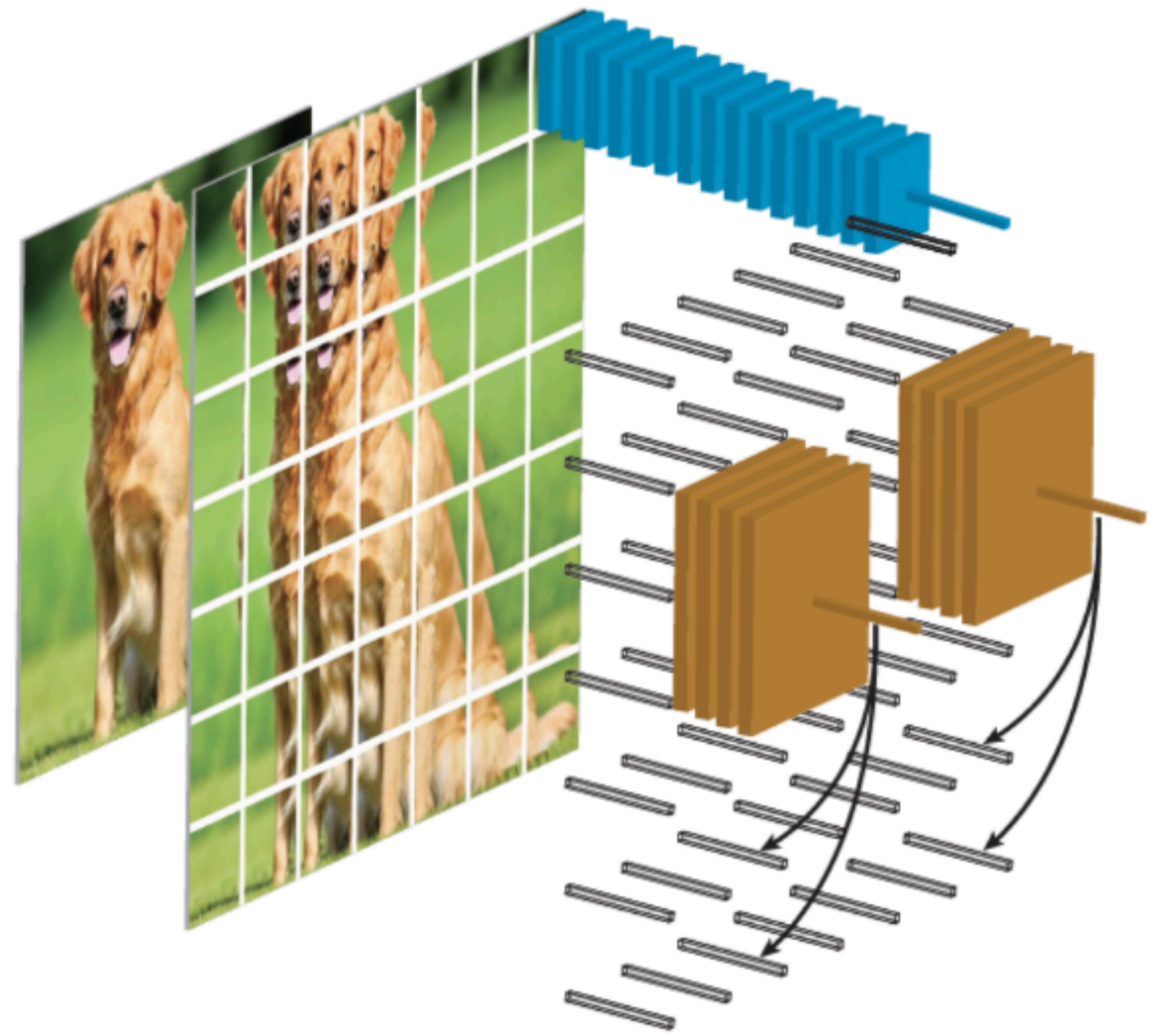
Contribution

- Propose a novel deep learning method for **local** self-supervised representation that **does not require labels nor end-to-end backpropagation** but exploits the natural order in data instead.
- Inspired by the observation that biological neural networks appear to learn without backpropagating a global error signal, we split a deep neural network into **a stack of gradient-isolated modules**
- Each module is trained to **maximally preserve the information** of its inputs using the InfoNCE bound from CPC.
- Each module improves upon the output of its predecessor, and that the representations created by the top module yield highly **competitive results** on downstream classification tasks in the audio and visual domain
- The proposal enables optimizing modules **asynchronously**, allowing large-scale distributed training of very deep neural networks on unlabelled datasets.

Contrastive Predictive Coding(09/11/2019)



CPC for Image



- An image is divided into a grid of overlapping patches.
- Each patch is encoded independently from the rest with a feature extractor (blue) which terminates with a mean-pooling operation, yielding a single feature vector for that patch.
- Doing so for all patches yields a field of such feature vectors (wireframe vectors).
- Feature vectors above a certain level (in this case, the center of the image) are then aggregated with a context network (brown), yielding a row of context vectors which are used to **linearly** predict (unseen) features vectors below.

CPC

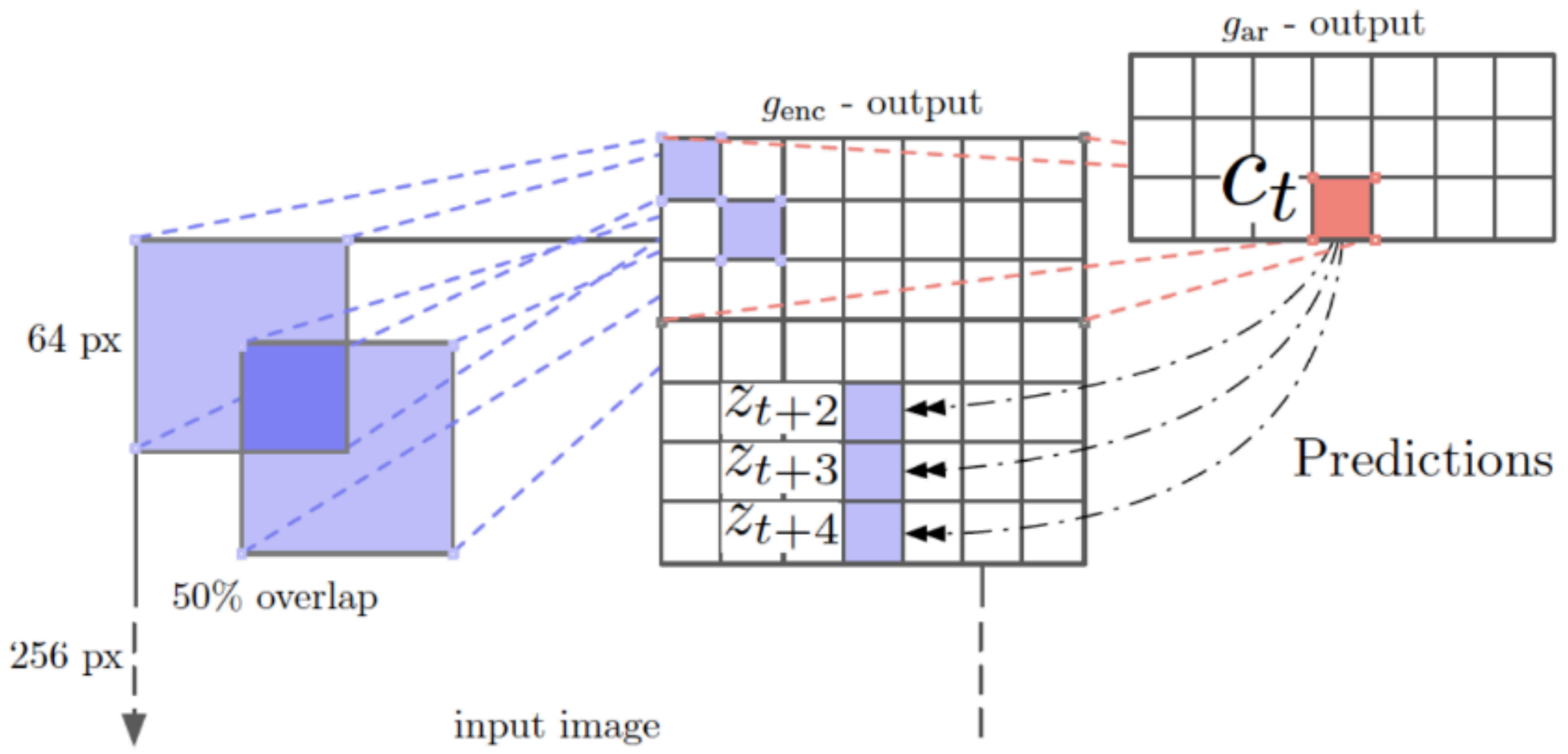


Figure 4: Visualization of Contrastive Predictive Coding for images (2D adaptation of Figure 1).

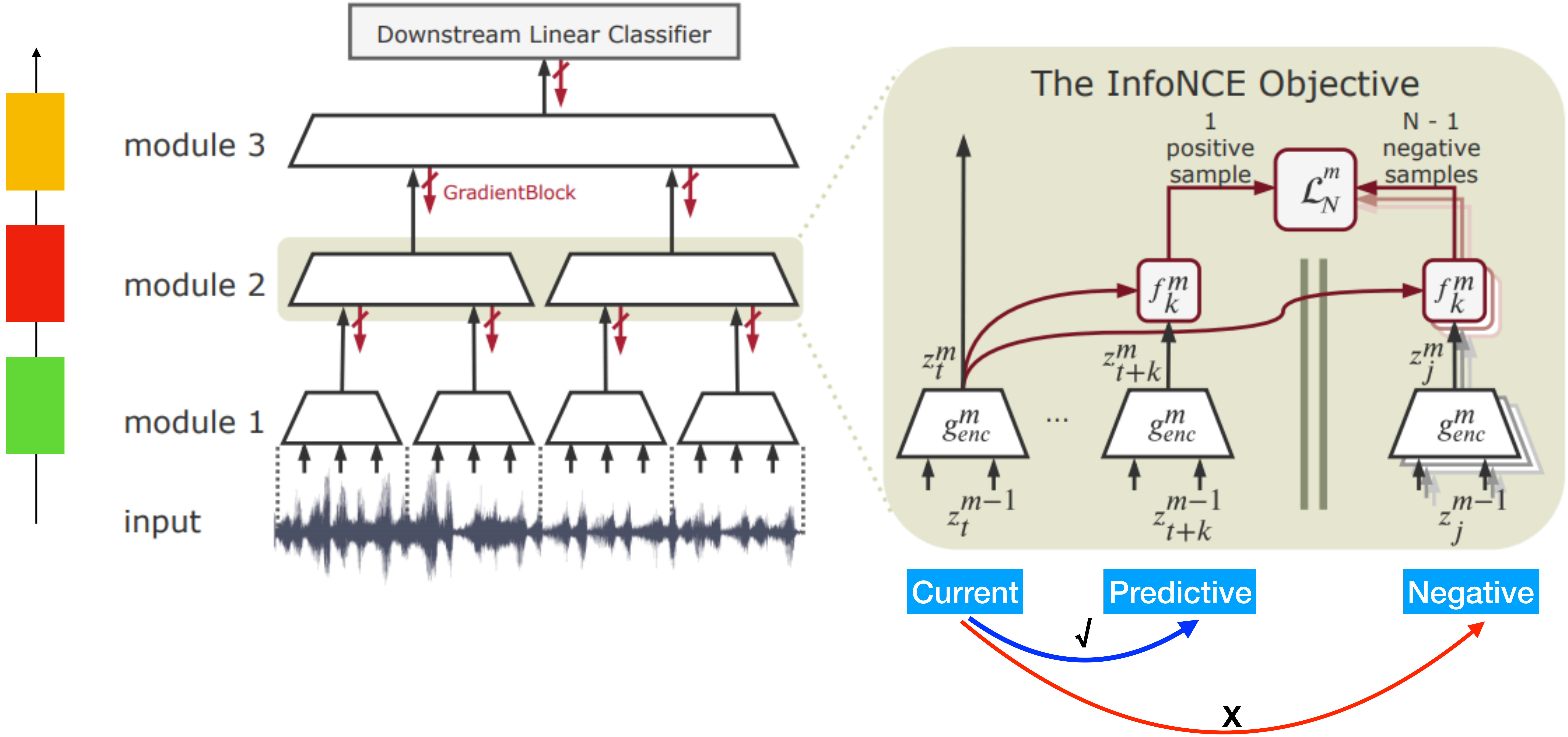
Prediction

$$\hat{z}_{i+k,j} = W_k c_{i,j}$$

Contrastive Loss

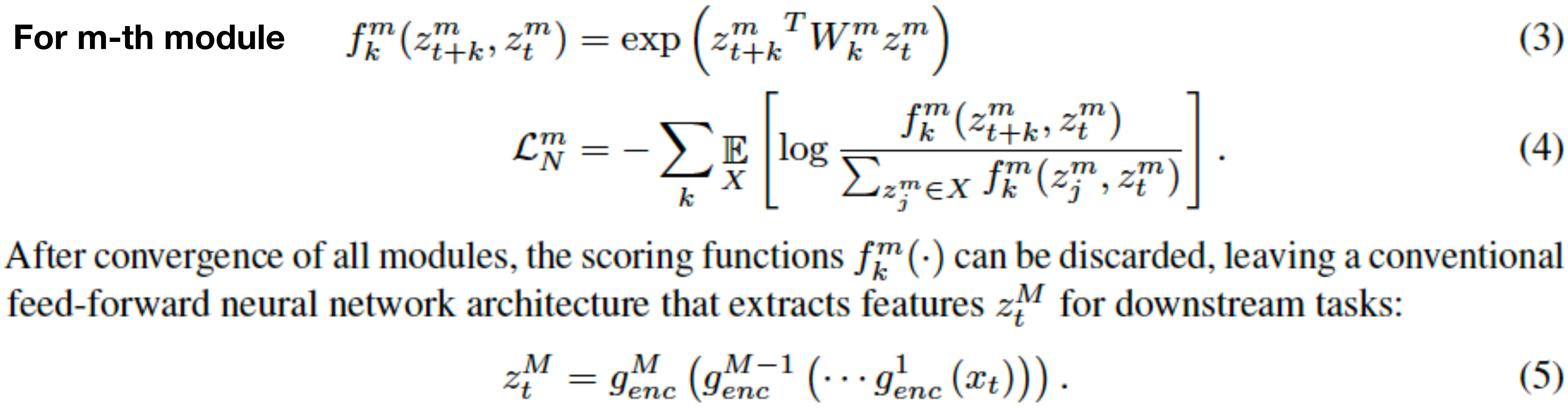
$$\begin{aligned}\mathcal{L}_{\text{CPC}} &= -\sum_{i,j,k} \log p(z_{i+k,j} | \hat{z}_{i+k,j}, \{z_l\}) \\ &= -\sum_{i,j,k} \log \frac{\exp(\hat{z}_{i+k,j}^T z_{i+k,j})}{\exp(\hat{z}_{i+k,j}^T z_{i+k,j}) + \sum_l \exp(\hat{z}_{i+k,j}^T z_l')}\end{aligned}$$

Greedy InfoMax (GIM)



InfoNCE Loss

For m-th module



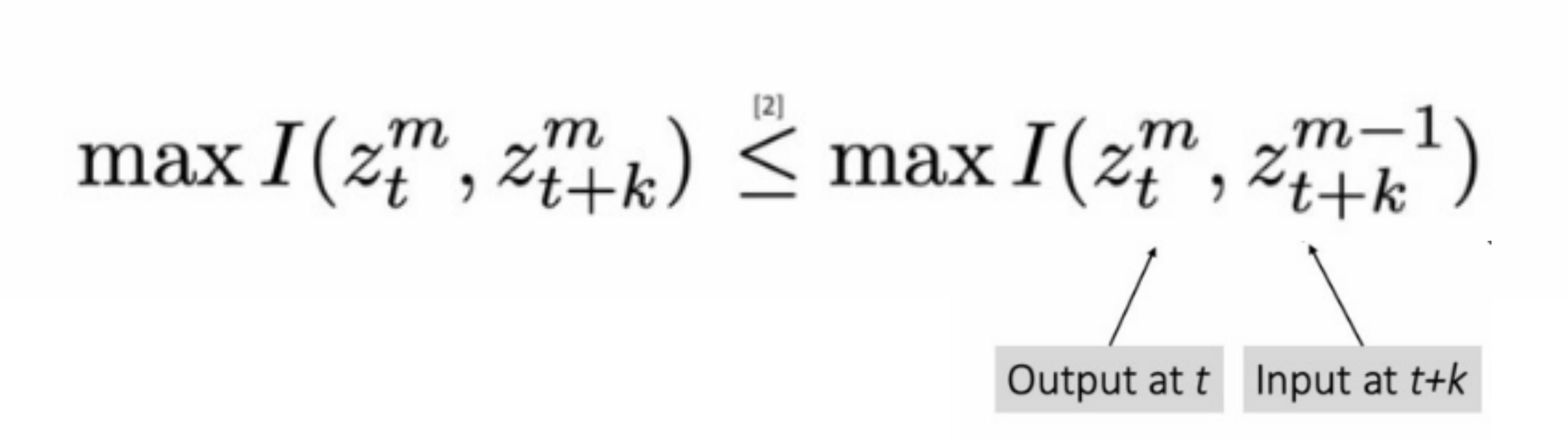
$$f_k^m(z_{t+k}^m, z_t^m) = \exp\left({z_{t+k}^m}^T W_k^m z_t^m\right) \quad (3)$$

$$\mathcal{L}_N^m = -\sum_k \mathbb{E}_X \left[\log \frac{f_k^m(z_{t+k}^m, z_t^m)}{\sum_{z_j^m \in X} f_k^m(z_j^m, z_t^m)}\right]. \quad (4)$$

After convergence of all modules, the scoring functions $f_k^m(\cdot)$ can be discarded, leaving a conventional feed-forward neural network architecture that extracts features z_t^M for downstream tasks:

$$z_t^M = g_{enc}^M\left(g_{enc}^{M-1}\left(\cdots g_{enc}^1\left(x_t\right)\right)\right). \quad (5)$$

InfoNCE Objective maximize mutual information between temporally nearby representations



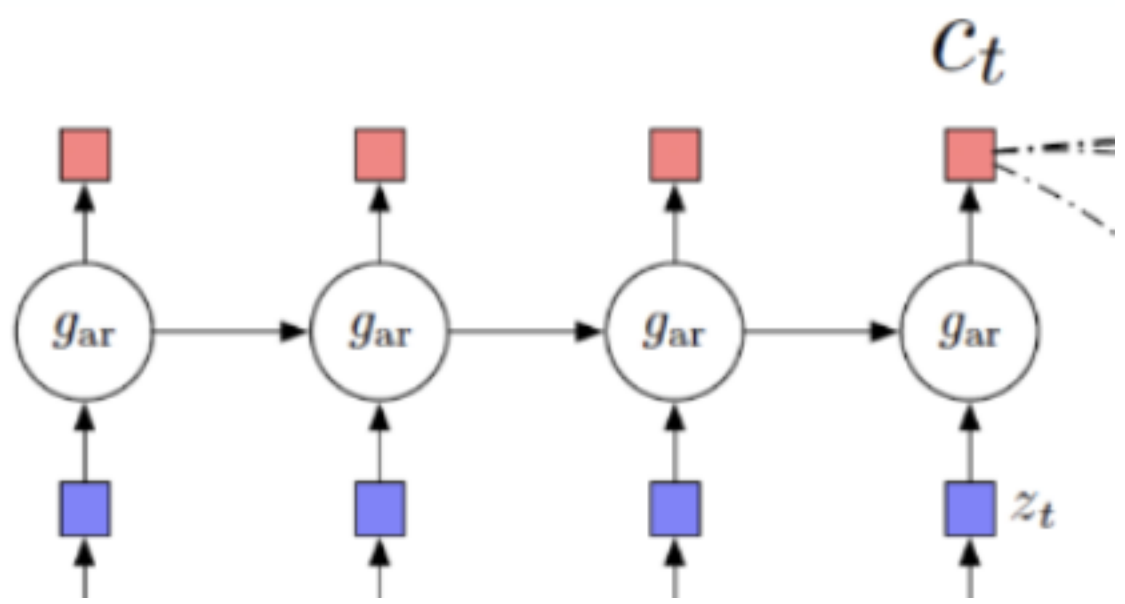
$$\max I(z_t^m, z_{t+k}^m) \overset{[2]}{\leq} \max I(z_t^m, z_{t+k}^{m-1})$$

Output at *t*
Input at *t+k*

Context-aggregate Representation

Context-aggregate representation

$$c_t^M = g_{ar}^M \left(\text{GradientBlock}\left(z_{0:t}^{M-1}\right)\right).$$



A GRU or PixelCNN-style model can serve in this role

$$f_k^M\left(z_{t+k}^{M-1}, c_t^M\right) = \exp\left(\text{GradientBlock}\left(z_{t+k}^{M-1}\right)^T W_k^M c_t^M\right).$$

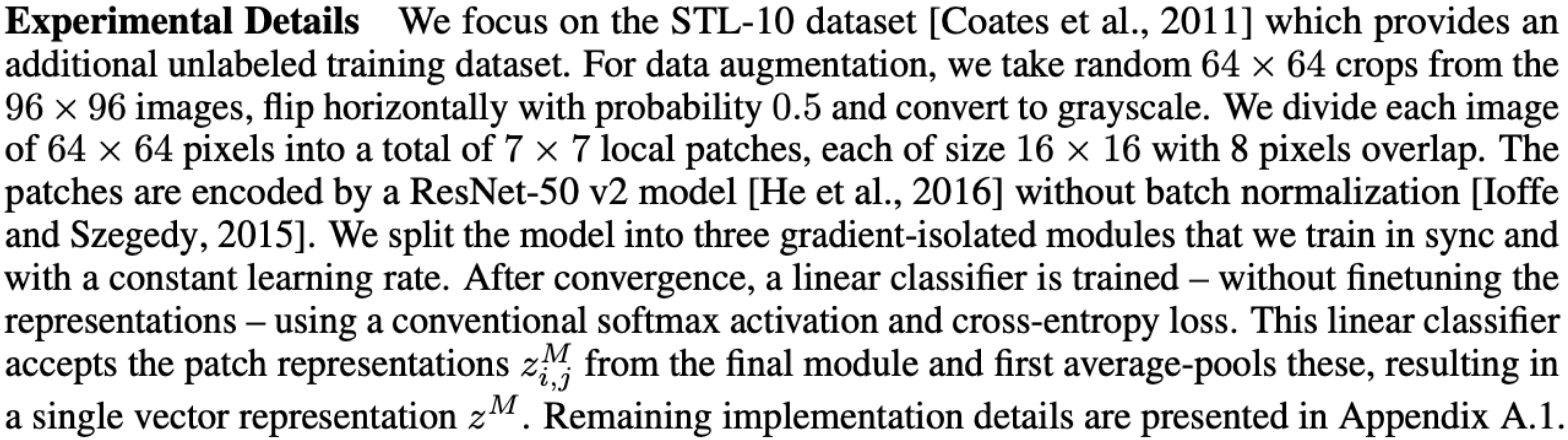
Practical Benefits

- **Applying GIM to high-dimensional inputs, we can optimize each module in sequence to decrease the memory costs during training.**
    - In the most memory-constrained scenario, individual modules can be trained, frozen, and their outputs stored as a dataset for next module.

- **GIM allows for training models on larger-than-memory input data with architectures that would exceed memory limitations.**
    - Leveraging the conventional pooling and strided layers found in common network architectures, we can start with small patches of the input, greedily train the first module, extract the now compressed representation spanning larger windows of the input and train the following module using these.

- **GIM provides a highly flexible framework for the training of neural networks.**
    - It enables the training of individual parts of an architecture at varying update frequencies. When a higher level of abstraction is needed, GIM allows for adding new modules on top at any moment of the optimization process without having to fine-tune previous results.

Vision Experiments



Experimental Details We focus on the STL-10 dataset [Coates et al., 2011] which provides an additional unlabeled training dataset. For data augmentation, we take random 64×64 crops from the 96×96 images, flip horizontally with probability 0.5 and convert to grayscale. We divide each image of 64×64 pixels into a total of 7×7 local patches, each of size 16×16 with 8 pixels overlap. The patches are encoded by a ResNet-50 v2 model [He et al., 2016] without batch normalization [Ioffe and Szegedy, 2015]. We split the model into three gradient-isolated modules that we train in sync and with a constant learning rate. After convergence, a linear classifier is trained – without finetuning the representations – using a conventional softmax activation and cross-entropy loss. This linear classifier accepts the patch representations $z_{i,j}^M$ from the final module and first average-pools these, resulting in a single vector representation z^M. Remaining implementation details are presented in Appendix A.1.

No autoregressive module is used for GIM

Result - Vision

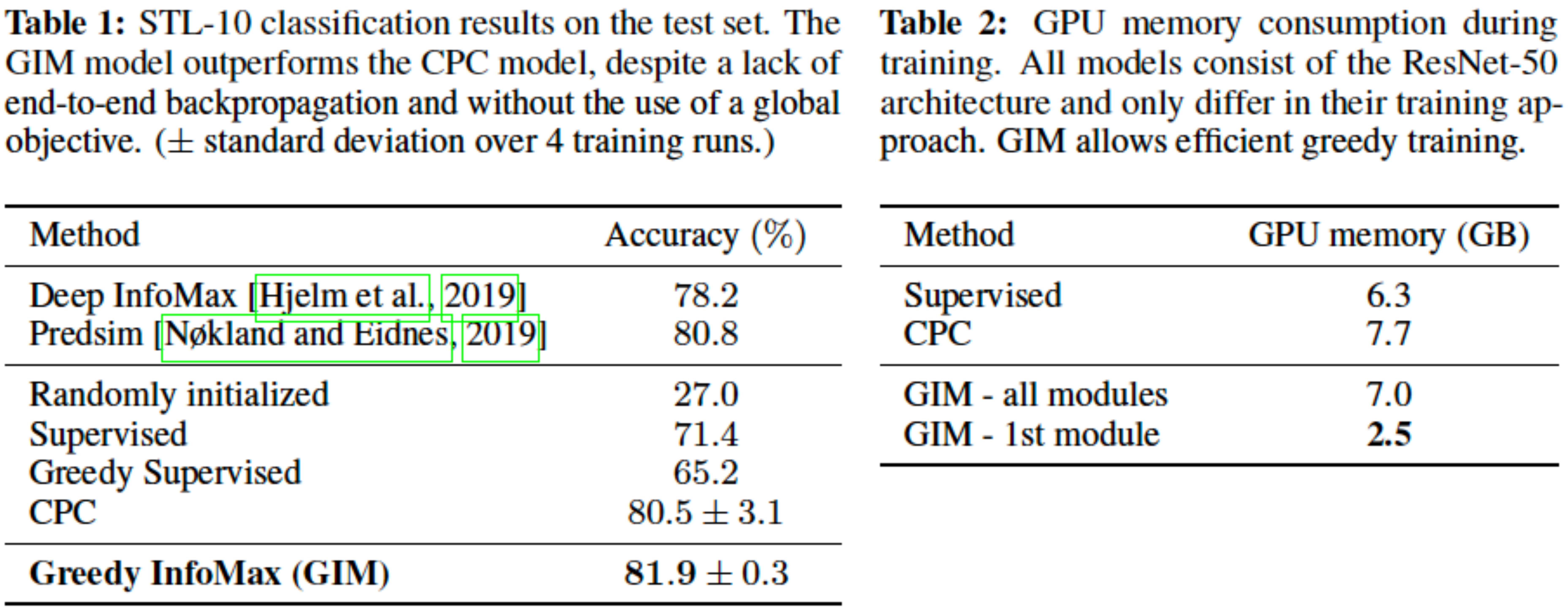


Table 1: STL-10 classification results on the test set. The GIM model outperforms the CPC model, despite a lack of end-to-end backpropagation and without the use of a global objective. (± standard deviation over 4 training runs.)

Method	Accuracy (%)
Deep InfoMax [Hjelm et al., 2019]	78.2
Predsim [Nøkland and Eidnes, 2019]	80.8
Randomly initialized	27.0
Supervised	71.4
Greedy Supervised	65.2
CPC	80.5 ± 3.1
Greedy InfoMax (GIM)	**81.9 ± 0.3**

Table 2: GPU memory consumption during training. All models consist of the ResNet-50 architecture and only differ in their training approach. GIM allows efficient greedy training.

Method	GPU memory (GB)
Supervised	6.3
CPC	7.7
GIM - all modules	7.0
GIM - 1st module	**2.5**

Semantic Features

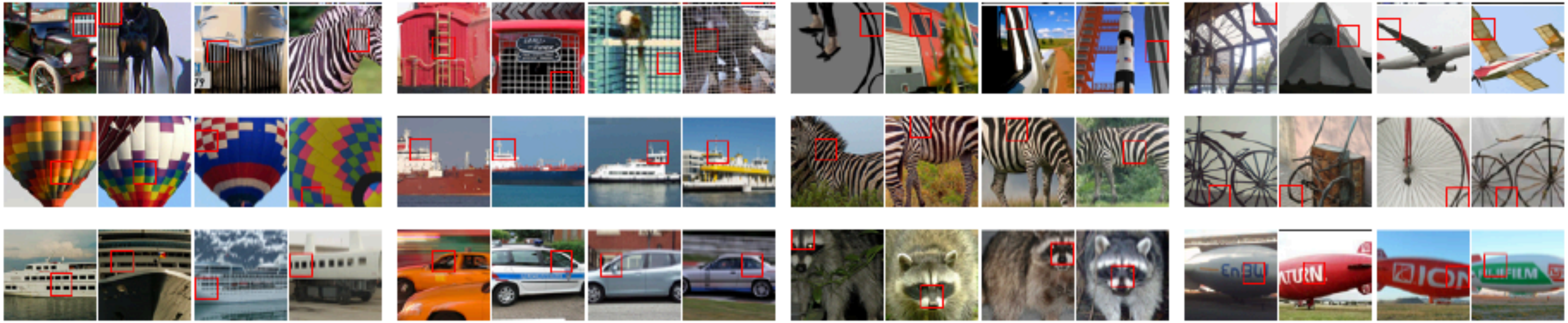
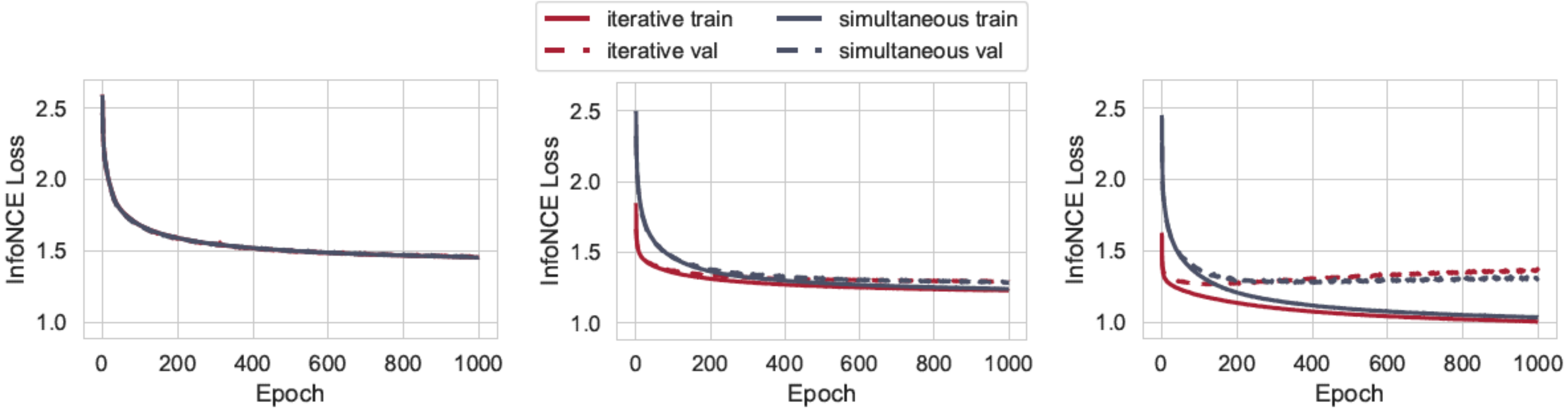


Figure 2: Groups of 4 image patches that excite a specific neuron, at *3* levels in the model (**rows**). Despite unsupervised greedy training, neurons appear to extract increasingly semantic features. Best viewed on screen.

We visualize patches that neurons in intermediate modules of the GIM model are sensitive to.
This demonstrates that modules later in the model focus on increasingly abstract features. Overall, the results demonstrate that complicated visual tasks can be approached using greedy self-supervised optimization, which can utilize large-scale unlabeled datasets.

Iterative vs Simultaneous



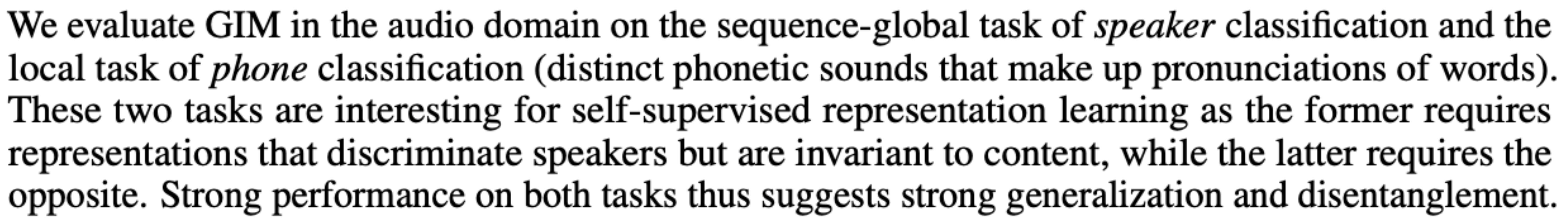
(a) First Module **(b)** Second Module **(c)** Third Module

Figure 3: Training curves for optimizing all modules *simultaneously* (blue) or *iteratively*, one at a time (red). While there is no difference in the training methods for the first module (**a**), later modules (**b, c**) start out with a lower loss and tend to overfit more when trained iteratively on top of already converged modules.

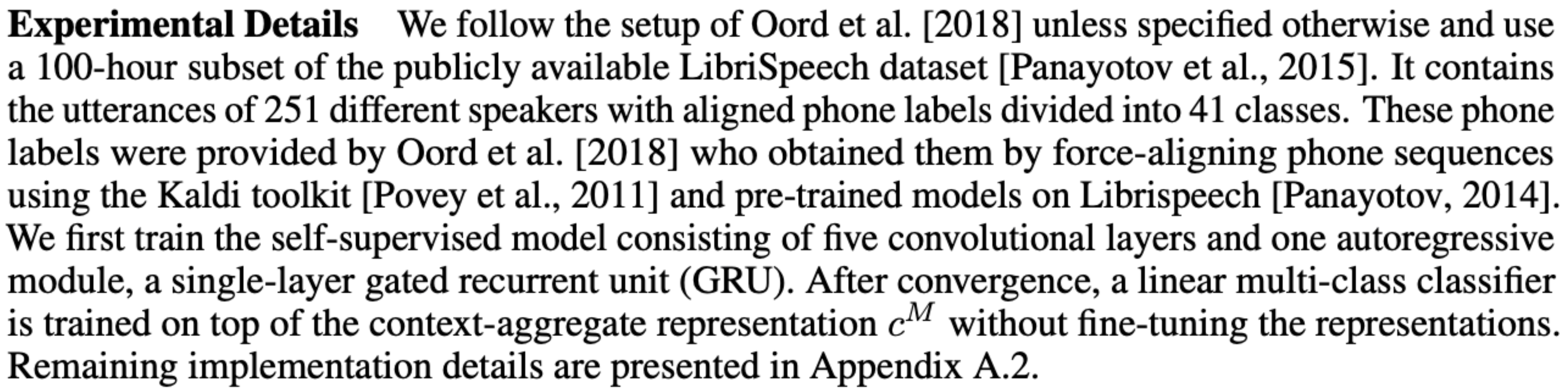
Overfitting: We tentatively attribute this to the regularizing effect from the initially noisy inputs received by the higher modules when training simultaneously

Audio Experiments

Two tasks



We evaluate GIM in the audio domain on the sequence-global task of *speaker* classification and the local task of *phone* classification (distinct phonetic sounds that make up pronunciations of words). These two tasks are interesting for self-supervised representation learning as the former requires representations that discriminate speakers but are invariant to content, while the latter requires the opposite. Strong performance on both tasks thus suggests strong generalization and disentanglement.



Experimental Details We follow the setup of Oord et al. [2018] unless specified otherwise and use a 100-hour subset of the publicly available LibriSpeech dataset [Panayotov et al., 2015]. It contains the utterances of 251 different speakers with aligned phone labels divided into 41 classes. These phone labels were provided by Oord et al. [2018] who obtained them by force-aligning phone sequences using the Kaldi toolkit [Povey et al., 2011] and pre-trained models on Librispeech [Panayotov, 2014]. We first train the self-supervised model consisting of five convolutional layers and one autoregressive module, a single-layer gated recurrent unit (GRU). After convergence, a linear multi-class classifier is trained on top of the context-aggregate representation c^M without fine-tuning the representations. Remaining implementation details are presented in Appendix A.2.

Results - Audio

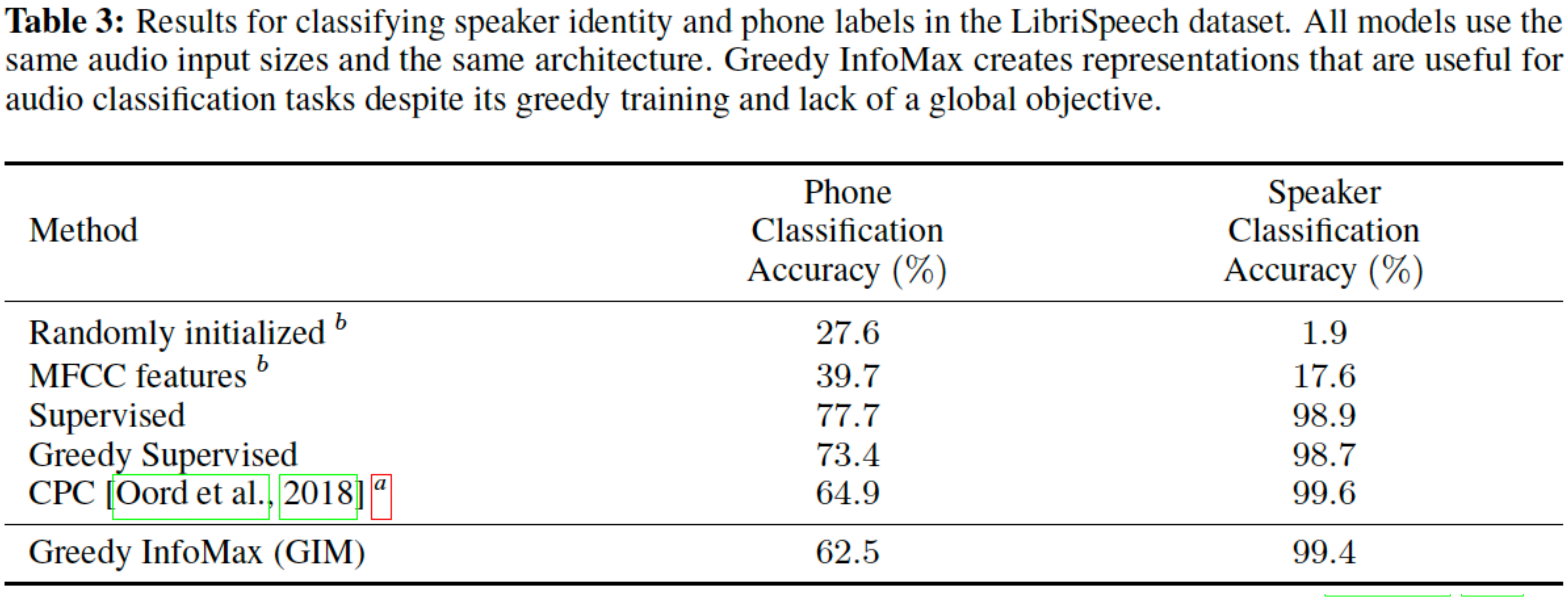


Table 3: Results for classifying speaker identity and phone labels in the LibriSpeech dataset. All models use the same audio input sizes and the same architecture. Greedy InfoMax creates representations that are useful for audio classification tasks despite its greedy training and lack of a global objective.

Method	Phone Classification Accuracy (%)	Speaker Classification Accuracy (%)
Randomly initialized [b]	27.6	1.9
MFCC features [b]	39.7	17.6
Supervised	77.7	98.9
Greedy Supervised	73.4	98.7
CPC [Oord et al., 2018] [a]	64.9	99.6
Greedy InfoMax (GIM)	62.5	99.4

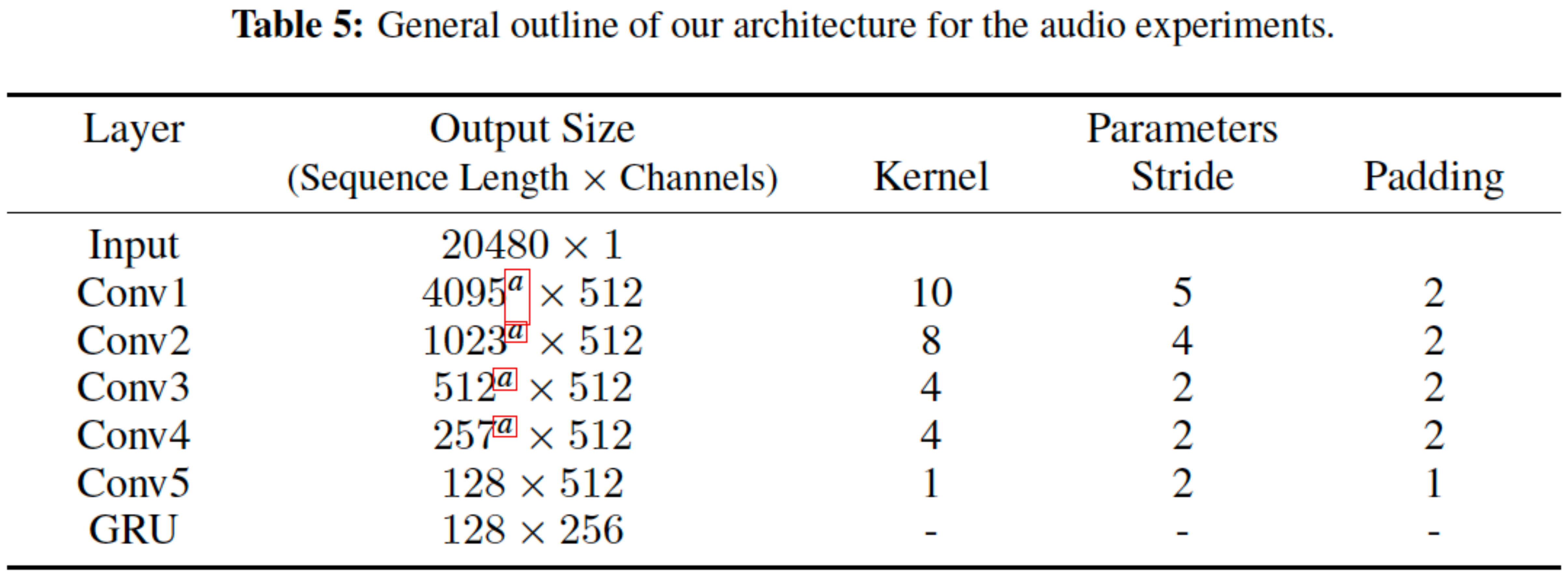
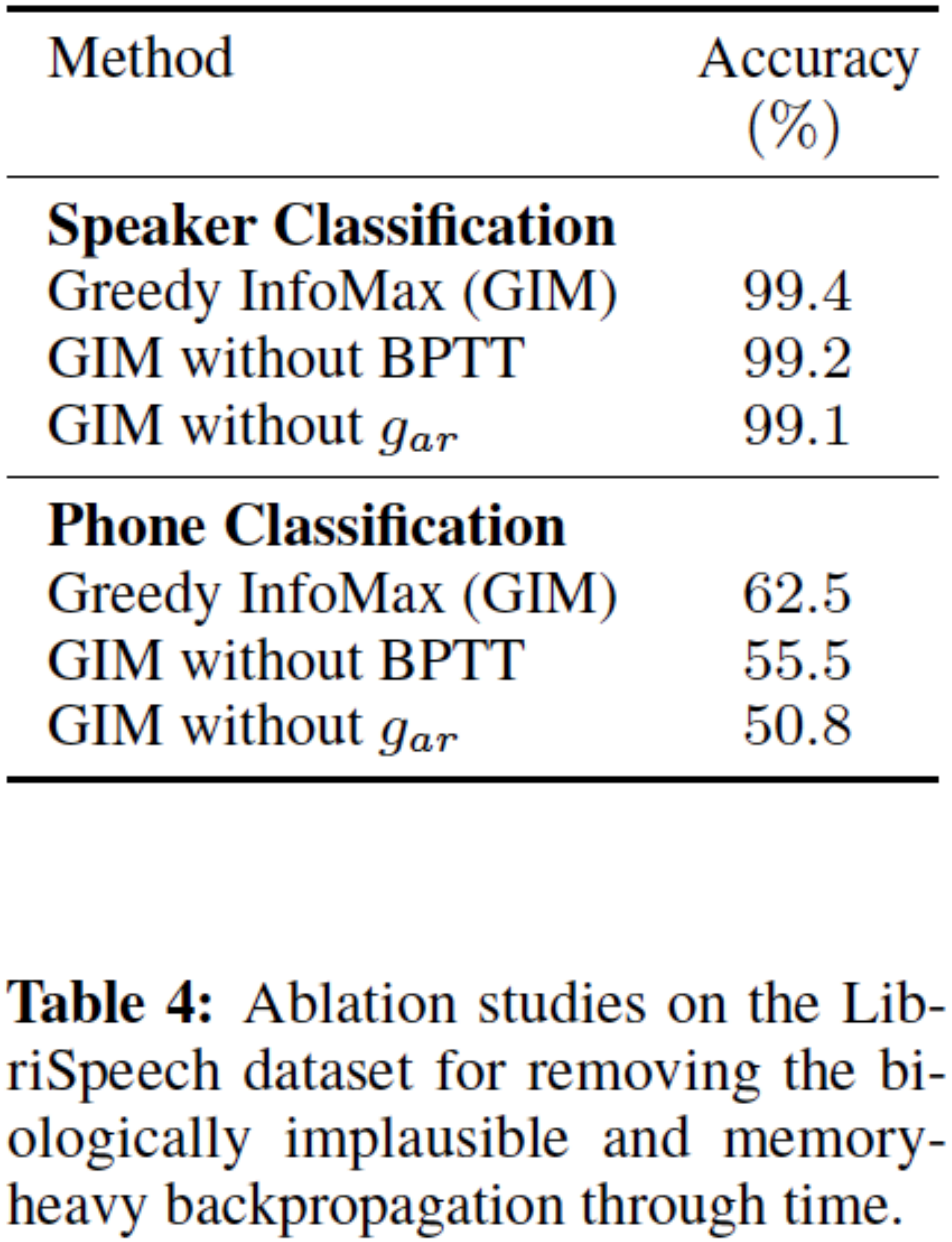


Table 5: General outline of our architecture for the audio experiments.

Layer	Output Size (Sequence Length $\times$ Channels)	Parameters Kernel	Stride	Padding
Input	20480×1			
Conv1	$4095^{a} \times 512$	10	5	2
Conv2	$1023^{a} \times 512$	8	4	2
Conv3	$512^{a} \times 512$	4	2	2
Conv4	$257^{a} \times 512$	4	2	2
Conv5	128×512	1	2	1
GRU	128×256	-	-	-

Overall, the discrepancy between better-than-supervised performance on the speaker task and less-than-optimal performance on the phone task suggests that
GIM and CPC are biased towards extracting sequence-global features.

Ablation study



Method	Accuracy (%)
Speaker Classification	
Greedy InfoMax (GIM)	99.4
GIM without BPTT	99.2
GIM without g_{ar}	99.1
Phone Classification	
Greedy InfoMax (GIM)	62.5
GIM without BPTT	55.5
GIM without g_{ar}	50.8

Table 4: Ablation studies on the LibriSpeech dataset for removing the biologically implausible and memory-heavy backpropagation through time.

Standard $c_t = g_{ar}(z_t, h_{t-1})$

GIM $c_t = g_{ar}(\text{GradientBlock}(z_t), h_{t-1})$

w/o BPTT $c_t = g_{ar}(\text{GradientBlock}(z_t), \text{GradientBlock}(h_{t-1}))$

The GIM approach performs best on downstream tasks where temporal or context dependencies do not need to be modeled by an autoregressive module. In these settings, GIM can outperform the CPC model, which makes use of end-to-end backpropagation, a global objective, and BPTT.

Ablation study

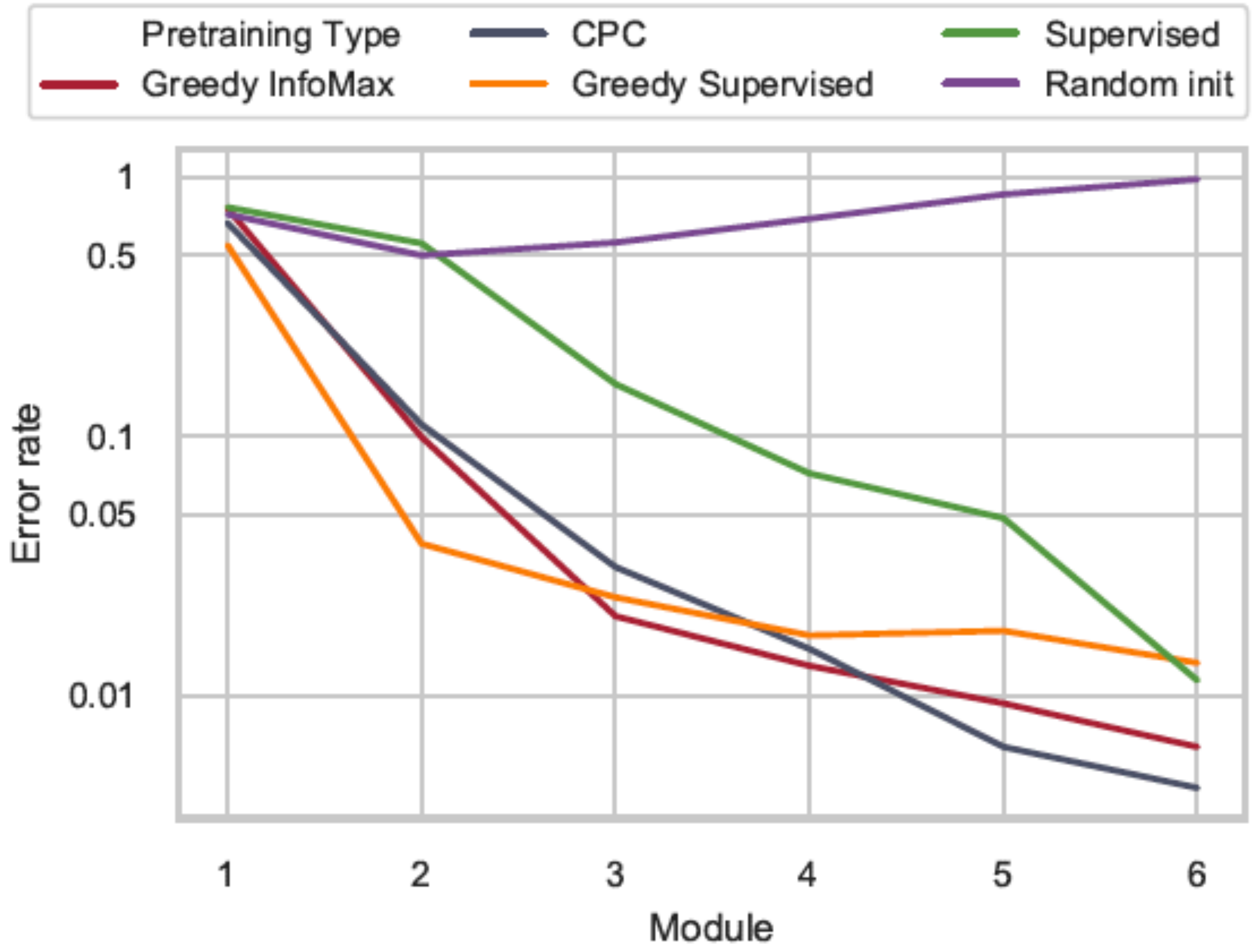


Figure 4: Speaker Classification error rates on a log scale (lower is better) for intermediate representations (layers 1 to 5), as well as for the final representation created by the autoregressive layer (corresponding to the results in Table 3).

This suggested that the InfoMax principle "stacks well", such that the greedy, iterative application of the InfoNCE loss performs similar to its global application.

Conclusion

- Presented Greedy InfoMax, a novel self-supervised greedy learning approach

- The relatively strong performance demonstrates that deep neural networks do not necessarily require end-to-end backpropagation of a supervised loss on perceptual tasks

- Our proposal enables greedy self-supervised training, which makes the model less vulnerable to overfitting, reduces the vanishing gradient problem and enables memory-efficient asynchronous distributed training