Smooth InfoMax - Towards Easier Post-Hoc Interpretability

Fabian Denoodt[1] (✉), Bart de Boer[2], and José Oramas[1]

[1] University of Antwerp, IDLab-imec, sqIRL
Prinsstraat 13, 2000 Antwerp, Belgium
fabian.denoodt@uantwerpen.be
[2] Vrije Universiteit Brussel
Pleinlaan 2, 1050 Brussels, Belgium

Abstract. We introduce Smooth InfoMax (SIM), a self-supervised representation learning method that incorporates interpretability constraints into the latent representations at different depths of the network. Based on β-VAEs, SIM's architecture consists of probabilistic modules optimized locally with the InfoNCE loss to produce Gaussian-distributed representations regularized toward the standard normal distribution. This creates smooth, well-defined, and better-disentangled latent spaces, enabling easier post-hoc analysis. Evaluated on speech data, SIM preserves the large-scale training benefits of Greedy InfoMax while improving the effectiveness of post-hoc interpretability methods across layers.
Our code is available via GitHub.

Keywords: Self-Supervised Representation Learning · Contrastive Learning · Post-Hoc Interpretability.

1 Introduction

Black-box models, particularly deep neural networks (NNs), have shown remarkable performance in recent years. However, despite their impressive success, their lack of interpretability poses a significant challenge, limiting their use in high-stakes decision environments. Consequently, various post-hoc interpretability techniques have been explored. Notable contributions include the work of [28], which aims to find the input image that maximally activates a specific neuron in the network, and the research by [33], which focuses on highlighting the regions in the input that a particular neuron is sensitive to.

However, the effectiveness of these post-hoc methods decreases in complex models due to the large number of neurons that must be analyzed. Additionally, as argued by [1], the internal semantic concepts learned by these neurons are typically highly entangled throughout the network. This makes the interpretation of a neuron particularly difficult, as multiple neurons may work as a whole and together be sensitive to a given semantic concept while other neurons may not be contributing anything at all. For these reasons, it is likely impossible to fully understand these NNs with just the existing post-hoc interpretability techniques.

In contrast, inherently interpretable models (e.g., logistic regression and decision trees) offer more transparency but may struggle with complex problems.

Another challenge with NNs is that they are typically trained end-to-end, which requires significant memory, especially as models grow larger. This can pose hardware constraints, as training must fit within the available memory of the device. Additionally, deeper networks can be more susceptible to the vanishing gradient problem [13].

To address these issues, we propose Smooth-InfoMax, a self-supervised representation learning method that integrates two existing paradigms: Greedy InfoMax (GIM) [18] and β-Variational Autoencoders (β-VAEs) [5]. This integration improves the post-hoc interpretability of the NN while enabling large-scale distributed training, combining benefits not achievable by either paradigm alone.

SIM's learning objective is based on contrastive learning and does not require labels or a decoder for training. Building upon GIM, SIM splits the architecture into modules, each trained greedily with a novel loss based on the InfoNCE bound [25]. As such, we preserve benefits such as large-scale distributed training of architectures that would otherwise not fit in memory and reduced vanishing gradients issues [18].

Furthermore, SIM incorporates the latent-space regularization properties of β-VAEs across various depths in the network. This helps create smooth and well-structured latent spaces that encourage disentanglement [5, 27, 12]. As a result, small changes in the latent space correspond to small changes in the input space, making post-hoc interpretability easier. However, unlike β-VAEs, SIM does not require a decoder during training, reducing memory usage. Another key difference is that SIM applies this regularization across different layers, making it easier to analyze representations throughout the network, rather than β-VAEs where the regularization typically is only applied at a single layer. A decoder can then be used as a post-hoc interpretability tool by traversing a latent space in the network, revealing the information that a particular neuron is sensitive to. Obtaining meaningful insights with such a procedure would be a lot harder if the spaces were not as well structured, as is typically the case in conventional NNs [7, 3].

Our contributions are the following:

1. Introducing SIM, a framework with a novel loss function and probabilistic architecture for easier interpretable latent spaces, evaluated on sequential speech data. Although a relatively straightforward integration of existing methods, this proposed combination provides specific benefits not achievable by either approach alone.
2. We show, via a decoder, that SIM produces latent spaces that are easier to analyze. This also leads to a new metric for quantifying the number of dimensions required for successful reconstructions.
3. Empirically showing that ideas from β-VAE extend to other frameworks and can be repeated at different depths without significant performance loss.

Reproducibility: Our code and commands to replicate the experiments are all available via GitHub.

2 The Starting Point - Greedy InfoMax

Greedy InfoMax (GIM) learns representations from sequential data without the need for labels by exploiting the assumption of slowly varying data [31]. This assumption is for instance applicable to speech signals where the conveyed information at time step t and $t+k$ contains redundancy, such as the speaker's identity, the conveyed emotion and the pronounced phonemes [18]. Meanwhile, this information may not necessarily be shared with random other patches of speech. An encoder can then be optimized to create representations that maximally preserve the shared information between the representations of temporally nearby patches [18], while at the same time discarding low-level information and noise that is more local [25]. It has been shown that such a strategy creates highly competitive representations for downstream tasks in various domains [11, 25, 18, 29, 24, 19, 4].

The network architecture An audio sequence is split up into patches $\mathbf{x}_1 \dots \mathbf{x}_T$ where each $\mathbf{x}_t$ is a vector of fixed length, containing for instance 10ms of speech. Each patch $\mathbf{x}_t$ is encoded by passing it through a series of M encoder modules: $g^1_{enc}(\cdot),\ g^2_{enc}(\cdot),\ \dots,\ g^M_{enc}(\cdot)$. An encoder module consists of one or more convolution layers. The final representation $\mathbf{z}^M_t$ is then obtained by propagating $\mathbf{x}_t$ through each module as follows:

$$g^M_{enc}(\dots g^2_{enc}(g^1_{enc}(\mathbf{x}_t))) = \mathbf{z}^M_t. \tag{1}$$

As such, each module's output is the input of the successive module: $g^m_{enc}(\mathbf{z}^{m-1}_t) = \mathbf{z}^m_t$. For tasks where context-related information is required, the final module g^M_{enc} can be replaced by an autoregressive module $g_{ar}(\mathbf{z}^{M-1}_1\ \dots\ \mathbf{z}^{M-1}_t) = \mathbf{c}_t$. The autoregressive module can for instance be represented as a Gated Recurrent Unit (GRU). Both $\mathbf{z}^M_t$ or $\mathbf{c}_t$ may serve as the representation for downstream tasks and can be pooled into a single vector if needed.

The loss function Given a representation $\mathbf{z}^m_t$ and a set $X = \{\mathbf{z}^m_1, \mathbf{z}^m_2, \dots\} \cup \left\{\mathbf{z}^m_{t+1}, \dots, \mathbf{z}^m_{t+K}\right\}$ consisting of random encoded audio patches and K subsequent samples of $\mathbf{z}^m_t$, respectively, GIM learns to preserve the information between temporally nearby representations by learning to discriminate the subsequent *positive* samples $\mathbf{z}^m_{t+k}$ from the *negative* random samples $\mathbf{z}^m_j$ using a function $f^m_k(\cdot)$ which scores the similarity between two latent representations [18]. This function is defined as follows:

$$f^m_k(\mathbf{z}^m_{t+k}, \mathbf{z}^m_t) = \exp({\mathbf{z}^m_{t+k}}^T W^m_k \mathbf{z}^m_t), \tag{2}$$

where W_k is a weight matrix which is learned. Intuitively, due to the slowly varying data assumption, the similarity score for positive patches should be high and small for negative patches. The InfoNCE loss, used to optimize an *individual* module $g^m_{enc}(\cdot)$ and its respective W^m_k is shown below:

$$\mathcal{L}^m_{\text{NCE}} = -\sum_k \mathop{\mathbb{E}}_{\text{X}} \left[\log \frac{f^m_k(\mathbf{z}^m_{t+k}, \mathbf{z}^m_t)}{\sum_{\mathbf{z}^m_j \in X} f^m_k(\mathbf{z}^m_j, \mathbf{z}^m_t)}\right]. \tag{3}$$

One can prove that minimizing the InfoNCE loss is equivalent to maximizing a lower bound on the mutual information between $\mathbf{z}_t^m$ and $\mathbf{z}_{t+k}^m$ [25]:

$$I(\mathbf{z}_{t+k}^m; \mathbf{z}_t^m) \geq \log(N) - \mathcal{L}_{\text{NCE}}^m. \tag{4}$$

As a result of GIM's greedy approach, a conventional neural network architecture can be divided into modules. These modules can be trained either in parallel on distributed devices or sequentially, enabling the training of models larger than device memory and reducing the vanishing gradient problem. In the following section, we discuss how we can preserve these benefits in SIM, while also allowing for better interpretability.

3 Smooth InfoMax

While optimizing for the InfoNCE bound, as done in GIM, is remarkably successful for downstream classification, analyzing the learned representations remains difficult. In what follows we introduce Smooth InfoMax (SIM), maintaining the computational benefits obtained from optimizing the InfoNCE objective, while introducing easily traversable latent spaces and better disentangled representations at different depths in the network due to techniques borrowed from β-VAEs.

3.1 Towards Decoupled Training for Probabilistic Representations

The architecture is again based on modules, where the modules $g_{enc}^1(\cdot)$, $g_{enc}^2(\cdot)$, $\ldots$, $g_{enc}^M(\cdot)$ are each greedily optimized without gradients flowing between them. However, rather than producing a single deterministic point $\mathbf{z}_t^m$, the output from $g_{enc}^m(\cdot)$ is now a multivariate Gaussian distribution $q(\mathbf{z}_t^m \mid \mathbf{z}_t^{m-1})$, parameterized by the mean vector $\boldsymbol{\mu}$ and covariance matrix $\text{diag}(\boldsymbol{\sigma})$. More precisely, we have:

$$g_{enc}^m(\mathbf{z}_t^{m-1}) = q(\mathbf{z}_t^m \mid \mathbf{z}_t^{m-1}) = \mathcal{N}(\boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma})), \tag{5}$$

with $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ dependent on $\mathbf{z}_t^{m-1}$. A point $\mathbf{z}_t^m$ is then obtained by sampling from this distribution, denoted respectively, as follows:

$$\mathbf{z}_t^m \sim q^m(\cdot \mid \mathbf{z}_t^{m-1}). \tag{6}$$

The encoding modules are thus stochastic and obtaining two representations from the same input will not necessarily produce the same result. This is in contrast to GIM's latent representations which remain fixed with respect to the input.

We obtain these stochastic modules by defining each module $g_{enc}^m(\cdot)$ consisting of two blocks. The first block receives as input $\mathbf{z}_t^{m-1}$ and predicts the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$. The second block samples $\mathbf{z}_t^m \sim q^m(\cdot \mid \mathbf{z}_t^{m-1})$ from this distribution and produces an output representation. In practice, sampling from q^m is achieved through a reparameterization trick, as introduced in [16]. The equation to compute $\mathbf{z}_t^m$ then becomes:

$$\mathbf{z}_t^m = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon},$$

where $\boldsymbol{\epsilon}$ corresponds to a sampled value $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\odot$ is element-wise multiplication. The two blocks are depicted in Figure 1. The optional autoregressive module $g_{ar}(\cdot)$ has been untouched, and remains identical as in GIM, resulting in deterministic representations.

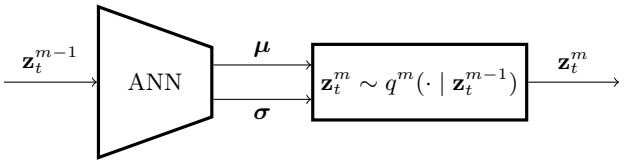

Fig. 1: A single module.

3.2 The Loss Function

Instead of training the NN's modules end-to-end with a global loss function, each module is optimized greedily with its own loss. Through the introduction of the *Smooth-InfoNCE* loss, mutual information between temporally nearby representations is maximized, while regularizing the latent space to be approximate to the standard Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$. This loss is defined as follows:

$$\mathcal{L}^m_{\text{S-NCE}} = \underbrace{-\sum_k \mathop{\mathbb{E}}_{\substack{\mathbf{z}^m_{t+k} \sim q^m(\cdot|\mathbf{z}^{m-1}_{t+k}) \\ \mathbf{z}^m_t \sim q^m(\cdot|\mathbf{z}^{m-1}_t)}} \left[\log \frac{f^m_k(\mathbf{z}^m_{t+k}, \mathbf{z}^m_t)}{\sum_{\mathbf{z}^m_j \in X} f^m_k(\mathbf{z}^m_j, \mathbf{z}^m_t)}\right]}_{\text{Maximize } I(\mathbf{z}^m_{t+k}, \mathbf{z}^m_t)} + \underbrace{\beta \; D_{KL}\left(q^m(\cdot \mid \mathbf{z}^{m-1}_t) \mid\mid \mathcal{N}(\mathbf{0}, \mathbf{I})\right)}_{\text{Regularisation}}. \quad (7)$$

Here, $m \in \mathbb{N}$ refers to the m'th module and $k \in \mathbb{N}$ the number of follow-up patches the similarity score $f^m_k(\mathbf{z}^m_{t+k}, \mathbf{z}^m_t)$ must rate. The latent representations $\mathbf{z}^m_{t+k}$ and $\mathbf{z}^m_t$ are encoded samples produced by $g^m_{enc}(\mathbf{z}^{m-1}_{t+k})$ and $g^m_{enc}(\mathbf{z}^{m-1}_t)$, respectively and X is a set of samples $\left\{\mathbf{z}^m_{t+k}, \mathbf{z}^m_1, \mathbf{z}^m_2, \dots\right\}$ where $\mathbf{z}^m_j$ with $j \neq t+k$ are random samples. In practice, the set can be based on the training batch. The parameter $\beta \geq 0$ is a hyper-parameter indicating the relative importance between the two terms. When $\beta = 0$, SIM is identical to GIM but with an altered architecture supporting probabilistic representations. The similarity score $f^m_k(\cdot)$ remains identical as in GIM:

$$f^m_k(\mathbf{z}^m_{t+k}, \mathbf{z}^m_t) = \exp(\mathbf{z}^m_{t+k}{}^T W^m_k \mathbf{z}^m_t). \quad (8)$$

$\mathcal{L}^m_{\text{S-NCE}}$ consists of two terms. The first term ensures that latent representations of temporally nearby patches maximally preserve their shared information. The second pushes the latent representations close to the origin.

The Gradient To estimate the expectation term in $\mathcal{L}_{\text{S-NCE}}$, we apply the same approximation method as in VAEs, achieved through Monte Carlo estimates [16].

The first term in $\mathcal{L}_{\text{S-NCE}}$ then becomes:

$$-\sum_{k} \frac{1}{L} \left[\sum_{l=1}^{L} \log \frac{f_k^m(\mathbf{z}_{t+k}^{m\,(l)}, \mathbf{z}_t^{m(l)})}{\sum_{\mathbf{z}_j^m \in X} f_k^m(\mathbf{z}_j^m, \mathbf{z}_t^{m(l)})} \right].$$

Here, L refers to the number of samples drawn. Each $\mathbf{z}_{t+k}^{m\,(l)}$ and $\mathbf{z}_t^{m(l)}$ are different samples produced by their respective distributions. However, similar to [16], we can set $L = 1$ without significantly hurting performance.

With regards to the second term in $\mathcal{L}_{\text{S-NCE}}$, since $q^m(\cdot \mid \mathbf{z}_t^{m-1})$ is a Gaussian defined by parameters $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$, a closed-form solution exists [16]. The closed-form equation is the following:

$$D_{KL}\left(q^m(\cdot \mid \mathbf{z}_t^{m-1}) \,||\, \mathcal{N}(\mathbf{0}, \mathbf{I})\right) = \frac{1}{2} \sum_{i=1}^{D} \left(-\log \sigma_i^2 - 1 + \sigma_i^2 + \mu_i^2\right).$$

The variable D refers to the number of dimensions of $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$. This term can thus, be directly computed and does not need to be approximated through Monte Carlo estimates. The gradient for the two terms can then be computed using automatic differentiation tools such as PyTorch.

3.3 Properties of the Latent Space

Here, we present two conjectures regarding the structure of the latent space defined by each of SIM's modules. They will serve as the main argument for why SIM's representations are more easily analyzable. Meanwhile, alternative contrastive approaches such as GIM lack these benefits.

Conjecture 1. *$\mathcal{L}_{S\text{-}NCE}$ enforces an uninterrupted and well-covered space around the origin.*
In SIM, a latent representation $\mathbf{z}_t^m \in \mathcal{Z}^m$ of a data point $\mathbf{z}_t^{m-1} \in \mathcal{Z}^{m-1}$ is a sample from a Gaussian distribution. Thus, encoding the same $\mathbf{z}_t^{m-1}$ an infinite number of times results in a spherical region (around a particular mean $\boldsymbol{\mu}$) in $\mathcal{Z}^m$ that is covered by the latent representations corresponding to $\mathbf{z}_t^{m-1}$, without any interruptions in this region. This is different from GIM where a data point merely covers a single point of the latent space (and not an entire region). Furthermore, because the KL divergence requires each region to be close to the origin, the regions are more likely to utilize the limited space efficiently around the origin, resulting in a lower chance of obtaining gaps between two regions from different data points.

Conjecture 2. *$\mathcal{L}_{S\text{-}NCE}$ enforces smooth and consistent transitions in the latent space with respect to the shared information between temporarily nearby patches.*
The argument on why this holds true is similar to the argument made for VAEs [16]. In the case of a VAE, a smooth space implies that a small change to $\mathbf{z}$ should result in a small change to its corresponding reconstruction, such that:

$$\mathbf{z} \approx \mathbf{z}' \implies p(\mathbf{x} \mid \mathbf{z}) \approx p(\mathbf{x} \mid \mathbf{z}'). \tag{9}$$

Indeed, one can observe that the KL-divergence will encourage the region of latent points that a data point $\mathbf{x}$ can map to to be large. Meanwhile, the reconstruction error in a VAE encourages all the latent points falling in this region to be as close as possible to the initial data point $\mathbf{x}$. In SIM, the same argument can be used to obtain:

$$\mathbf{z}_t^m \approx \mathbf{z}_t^{m\prime} \implies f(\mathbf{z}_{t+k}^m, \mathbf{z}_t^m) \approx f(\mathbf{z}_{t+k}^m, \mathbf{z}_t^{m\prime}), \tag{10}$$

resulting in a smooth space with respect to the shared information between temporally nearby patches. Additionally, if a decoder is trained on SIM's representations, for the same reason, we obtain:

$$p(\mathbf{x}_t \mid \mathbf{z}_t^m) \approx p(\mathbf{x}_t \mid \mathbf{z}_t^{m\prime}). \tag{11}$$

Traversability of the space As a result of the smooth and well-defined shape, one can make small changes to $\mathbf{z}_t^m$ and observe what happens through a decoder with a much smaller risk of having abrupt changes to the corresponding $\mathbf{x}$, or obtaining out-of-distribution latent points that correspond to non-meaningful reconstructions due to gaps in the latent space. This results in an easily traversable latent space with a predictable structure, which is not guaranteed in conventional NNs, as they typically do not enforce these additional constraints.

Disentanglement GIM poses no direct constraints on disentanglement risking having many dimensions of the representation together contribute a small amount to the contained information of an individual concept. However, as argued by [12], setting the prior $p(\mathbf{z})$ of the β-VAE's loss to an isotropic Gaussian encourages disentanglement in the representations. This results in each dimension from the encoding to capture a different property of the original data. In the case of $\mathcal{L}_{\text{S-NCE}}$, the prior corresponds to the standard normal $\mathcal{N}(\mathbf{0}, \mathbf{I})$, and thus, this theorem is also applicable to SIM, and choosing a large value for β in $\mathcal{L}_{\text{S-NCE}}$ applies more pressure for the representations to be better disentangled.

4 Experiments and Evaluation

We evaluate SIM's latent representations on raw speech data and compare them against GIM as a baseline. The goal is to measure the impact of β-VAE regularization in terms of raw performance and to analyze how effectively post-hoc interpretability techniques would perform. Since this work uses the well-established properties of β-VAEs, we do not aim to independently revalidate them. β-VAE's disentanglement, in particular, has been extensively confirmed [15, 12, 6] and benchmarked against other methods in prior work [17].

4.1 Setup

Dataset We use two publicly available speech datasets. The first is an artificial dataset[3] with a known and predictable structure, chosen because it provides a

[3] Available at `https://github.com/fdenoodt/Artificial-Speech-Dataset`.

clear expectation of what a learning system should capture. The second, following GIM, is the 100-hour subset of the large-scale LibriSpeech dataset [18, 26]. The artificial dataset contains 851 fixed-length (640 ms) audio files, sampled at 16 kHz and split into 80% training and 20% test sets. Each file consists of a single spoken sound composed of alternating consonants and vowels (e.g., "gi-ga-bu").

Architecture SIM's architecture consists of three probabilistic CNN-based encoder modules and one autoregressive GRU module. Each CNN layer in these modules has 512 hidden dimensions. In g^1_{enc}, two convolutions (kernel: 10, 8; stride: 5, 4; padding: 2) are followed by parallel μ and σ convolution layers (kernel: 1, stride: 1, no padding). g^2_{enc} contains two convolutions (kernel: 4; stride: 2; padding: 2), followed by μ and σ convolutions (kernel: 1, no padding). g^3_{enc} has one convolution (kernel: 4, stride: 2, padding: 1) followed by μ and σ convolutions (kernel: 1, no padding). The final module, g_{ar}, is a GRU with an output of size 64×256. ReLU is applied after each convolution except in the μ and σ layers. The total downsampling factor is 160, producing a feature vector for every 10 ms of speech. Batch norm is applied to LibriSpeech but not to the artificial dataset. While it wasn't strictly necessary for LibriSpeech, it significantly increased training speed, which is beneficial given the dataset's large size. All modules are trained in parallel without gradients flowing between modules.

Training Procedure SIM is trained with the Adam optimizer (learning rate: 2×10^{-4}, batch size: 8). The maximum number of patches to predict in the future K is set to 10, with 1000 epochs on the artificial dataset and 100 on LibriSpeech. The regularization weight β is set to 0.01 on the artificial dataset to encourage interpretability and 0.001 on LibriSpeech to balance interpretability and performance. Implementation details regarding drawing negative samples for $f^m_k(\cdot)$ remain identical to the audio experiment from [18].

4.2 Classification Performance

We evaluate SIM's representations by training a fully connected linear layer on top of SIM's frozen pretrained backbone for classification tasks. Classifiers are trained on temporally-average-pooled representations for 10 epochs using Cross-Entropy and the Adam optimizer ($lr = 0.001$). Tasks include vowel (3 labels) and syllable (9 labels) classification on the artificial dataset, and phoneme (41 labels) and speaker (251 labels) classification on LibriSpeech.

Results. A known drawback of β-VAE regularization is increased performance degradation, as greater emphasis is placed on disentanglement through the hyperparameter β [15]. Interestingly, Table 1 shows that this trade-off is quite manageable, especially considering that SIM applies this regularizer across different layers. Both SIM and CPC achieve high accuracy on speaker (96.02%, 98.60%) and vowel (92.58%, 95.24%) classification but perform worse on syllable (44.53%, 50.00%) and phoneme (60.22%, 61.80%) classification. This suggests that the InfoNCE objective favors global sequence features while preserving less

Table 1: Accuracy for classification tasks on the artificial speech and LibriSpeech datasets. [a]Baseline results from [18].

Artificial Speech Dataset		
Method	Vowel Classification (%)	Syllable Classification (%)
Supervised	91.19 ± 1.56	83.32 ± 2.06
Random Backbone	32.44 ± 4.44	9.88 ± 2.12
GIM	95.24 ± 0.60	50.0 ± 1.55
SIM	92.58 ± 2.06	44.53 ± 1.77
LibriSpeech Dataset		
Method	Speaker Classification (%)	Phone Classification (%)
Supervised[a]	98.90	77.70
Random Backbone[a]	1.90	27.60
GIM	98.60	61.80
SIM	96.02	60.22

local information. Adding a hidden layer improved training accuracy for the syllable task but did not improve test performance, indicating that consonant information may no longer be fully retained in the representations. Meanwhile, the randomly initialized backbone performs poorly across all tasks, confirming that SIM learns meaningful representations.

4.3 Representation Analysis

Qualitative Assessment of Latent Space Smoothness To gain a notion of the smoothness of SIM's latent space, we train a decoder $D(\mathbf{z}_t^3) = \widetilde{\mathbf{x}}_t$ on the artificial dataset, using representations from $g_{enc}^3(\cdot)$ to decode interpolations between two latent representations. Two audio signals, "bidi" and "baga" are encoded into their respective latent representations, $\mathbf{z}_{\text{bidi}}^3$ and $\mathbf{z}_{\text{baga}}^3$ (64×512). Interpolated representations $\mathbf{z}_\alpha^3 = (1-\alpha)\mathbf{z}_{\text{bidi}}^3 + \alpha \mathbf{z}^3{}_{\text{baga}}$ are decoded for values of α between 0 and 1.
Results. Fig. 2 shows the decoded signals smoothly transitioning as α varies, with no abrupt changes or nonsensical outputs. Exploring other interpolated audio signals than the one presented here, is possible via our demo. When decoding real samples (non-interpolated), we noted that vowel sounds were consistently correct, but consonants were often unclear or incorrect, which aligns with our discussion in 4.2 that consonant information may be less represented.

Quantitative Evaluation of Specific Information Spread To assess how vowel and speaker information is distributed across latent dimensions, we train linear classifiers (without bias) on average-pooled representations from $g_{enc}^1(\cdot)$, $g_{enc}^2(\cdot)$, and $g_{enc}^3(\cdot)$. Classifier weights indicate the relevance of each dimension, with large magnitudes signifying high importance.
Results. As shown in Fig. 3, SIM concentrates vowel/speaker information in fewer dimensions, which is beneficial for interpretability. GIM, on the other hand, spreads this information more broadly thereby requiring a larger number of

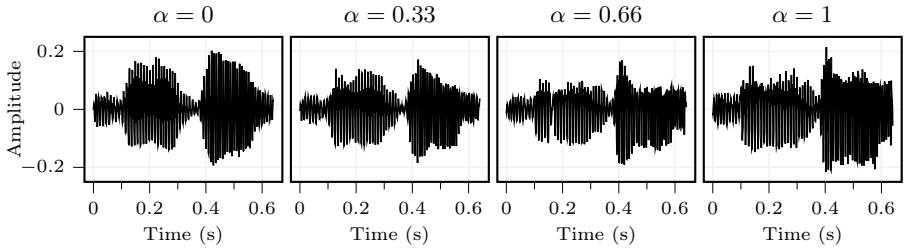

Fig. 2: Interpolated latent representations between two audio signals ("bidi" and "baga") using SIM. Each plot shows the decoded signal for different interpolation factors α. Listen to the decoded audio, among others, on Google Colab.

neurons to be studied. Accuracy for vowel identification in GIM: 95.94%, 92.81%, 94.06%, and in SIM: 87.5%, 93.44%, 91.87%. For speaker identification, GIM: 88.29%, 97.83%, 98.56%, and SIM: 68.36%, 91.76%, 94.32%.

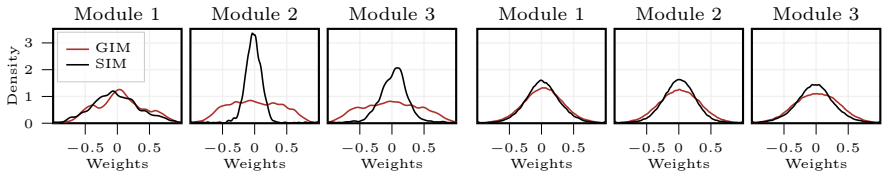

Fig. 3: Distribution of linear classifier weights for vowel prediction (artificial dataset, left) and speaker identification (LibriSpeech, right), trained on representations from $g^1_{enc}(\cdot), g^2_{enc}(\cdot), g^3_{enc}(\cdot)$. SIM's classifiers show more weights near zero, indicating that vowel and speaker information is concentrated in fewer dimensions.

Quantitative Evaluation of General Information Spread To further observe the impact on interpretability through unit analysis and to analyze whether our representations align with the known disentanglement properties from β-VAE's regularizer [12, 6, 15], we introduce a metric to measure how effectively a decoder $D : \mathcal{Z}^m \to \mathcal{X}$ can reconstruct a target signal when only the most relevant latent dimensions are modified. This serves as a proxy for entanglement, especially given that the artificial dataset has only a few ground truth factors, requiring far fewer dimensions than the available 512.

For each pair of starting and target representations $\left(\mathbf{z}^m_{\text{start}}, \mathbf{z}^m_{\text{target}}\right)$, we define an interpolated representation, $\mathbf{z}^m_{\alpha=1}$, where only the N most important dimensions (those with the greatest average difference over 64 time frames) are altered to match $\mathbf{z}^m_{\text{target}}$. The similarity between the decoded $\mathbf{z}^m_{\alpha=1}$ and $\mathbf{z}^m_{\text{target}}$ measures

how many dimensions are needed to transform the signal. The relative error is computed as:

$$\delta = \frac{\text{MAE}\left(D(\mathbf{z}^m_{\text{target}}), D(\mathbf{z}^m_{\alpha=1})\right)}{\text{MAE}\left(D(\mathbf{z}^m_{\text{start}}), D(\mathbf{z}^m_{\alpha=1})\right)} \tag{12}$$

Here, δ ranges from 0 (exact match to target) to 1 (no effect from altering dimensions).

Decoder Details. Each decoder $D(\mathbf{z}^m_t)$ is trained for 50 epochs on representations from a specific module $g^m_{enc}(\cdot)$, using the MSE loss, a learning rate of 2×10^{-4}, and batch size 8. The decoder mirrors the encoder architecture, but for the first module, two additional layers (kernel size 3, padding 1, stride 1) were added to improve reconstruction.

Results. Table 2 shows the relative errors. Across all depths, SIM reconstructs the target signal using fewer dimensions than GIM. For the artificial dataset, GIM needs at least half of the dimensions for successful reconstruction, whereas SIM achieves similar results with just 1/8th. Given the limited information in this dataset, which theoretically requires far fewer than 512 dimensions, multiple of GIM's dimensions seem sensitive to similar attributes. This implies a more entangled representation, which aligns with earlier findings in Fig. 3. For LibriSpeech, SIM consistently requires fewer dimensions, averaging around half the number used by GIM, showing potential to scale well to more complex datasets.

Table 2: Relative reconstruction error δ (%) when only the N most important dimensions out of 512 are active. Lower values are preferred. GIM distributes relevant information across more dimensions than SIM.

Module	Method	Artificial Speech Dataset									LibriSpeech Dataset								
		2	**4**	**8**	**16**	**32**	**64**	**128**	**256**	**512**	**2**	**4**	**8**	**16**	**32**	**64**	**128**	**256**	**512**
g^1_{enc}	GIM	99.32	98.71	97.6	95.46	91.2	82.35	62.96	24.33	0	95.55	91.77	87.47	81.12	71.65	58.02	39.52	16.8	0
	SIM	84.01	62.5	37.42	29.45	23.64	17.96	11.97	5.66	0	88.02	79.83	64.92	36.93	15.55	10.62	6.96	3.53	0
g^2_{enc}	GIM	98.86	97.91	96.17	93.07	87.58	77.8	59.81	27.46	0	97.35	94.88	90.82	84.7	76.6	67.02	54.25	31.32	0
	SIM	89.89	81.78	65.96	39.26	28.04	20.98	14.0	6.97	0	88.75	82.32	74.74	65.83	49.4	32.47	22.44	12.39	0
g^3_{enc}	GIM	99.01	98.16	96.59	93.9	89.29	81.09	65.23	31.76	0	94.37	90.65	85.08	78.1	71.47	65.68	57.0	35.95	0
	SIM	92.74	87.44	77.56	59.27	39.89	27.96	17.55	7.73	0	86.76	81.14	75.54	70.96	64.75	52.02	39.23	23.38	0

Qualitative Analysis of Latent Shape To evaluate the structure of SIM's latent space, we first compute the representations $\mathbf{z}^3_t = g^3_{enc}(\mathbf{x}_t)$ for each sample from the test set. For each of their 512 dimensions, we construct a histogram with 100 bins, showing how all activations of the dataset for an individual dimension are spread. Figure 4 displays the histograms for a selected set of dimensions. SIM's activations consistently follow a Gaussian distribution, aligning with our design goal of regularizing the latent space toward a standard normal distribution. This predictable structure helps post-hoc interpretability tools by clearly describing the regions of interest. In contrast, the latent representations

produced by GIM show greater variation. In particular, dimension 9 for the artificial dataset, and dimensions 13 and 14 for LibriSpeech are shifted away from the origin or differ from the other dimensions. Note that in GIM's current implementation, the latent spaces are implicitly constrained to center around the origin due to the use of a bias-free discriminator in Eq. 2. If the discriminator were non-linear or had a bias term, the shape of the latent space could potentially be even less predictable.

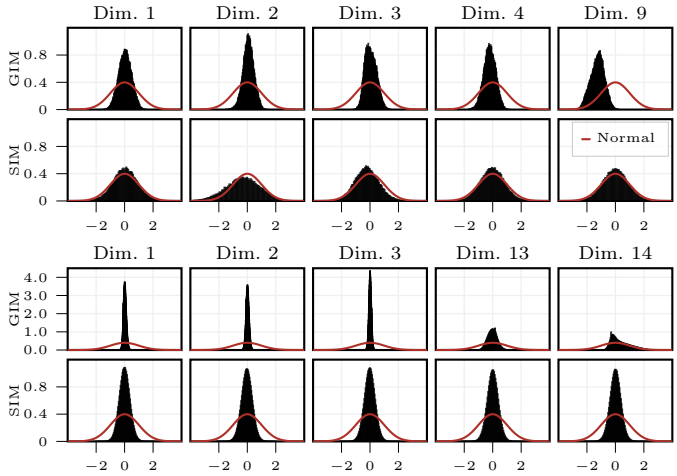

Fig. 4: Top rows: Artificial Dataset, bottom: LibriSpeech. Distribution of activations per dimension. SIM's activations have a consistent shape across dimensions, whereas GIM enforces no latent space constraints, resulting in greater variation in certain dimensions. Other dimensions are available here.

5 Related Work

This work uses the benefits of β-VAE's regularization, applying them across multiple layers to improve post-hoc analysis of the network.

In terms of *interpretability through regularization*, various approaches have been explored. For instance, sparsity regularization in the activations of hidden representations [30, 2, 10] has been shown to improve the compression of information into fewer dimensions, reducing the number of neurons to analyze, yet this does not encourage disentanglement or smoothness. Similarly, [22] improves model interpretability through regularization of activations, modifying the model's behavior such that existing explainability methods produce explanations that better align with human perception. Another notable method involves tree regularization [32], which constrains neural networks to be well-

approximated by decision trees, thereby improving interpretability. While effective, its applications are typically limited to simpler tasks where decision trees can easily be formed. In contrast, SIM relies on post-hoc tools for its analysis but is able to model complex audio or vision tasks. Regarding disentanglement, traditional methods have typically focused on regularizing a single layer in the architecture [12, 6, 15, 14, 9], whereas SIM applies one of these approaches in a new context and across multiple layers.

For *post-hoc interpretability*, SIM's decoder is similar to approaches like [8, 28], which reconstruct inputs from latent representations but rely on gradient ascent rather than a learned decoder. However, since NNs are not typically bijective, the reconstructions found do not necessarily map to a similar point from the dataset, resulting in often noisy and unclear reconstructions [8, 28]. To improve intelligibility, recent feature-activation-based methods incorporate human priors [23, 21, 20], guiding reconstructions toward more interpretable outputs. Decoders offer an alternative by directly learning to reconstruct inputs, implicitly encoding priors from the training data. However, they may introduce hallucinations that do not fully reflect the original model. Nonetheless, both gradient-based and decoder-based approaches could benefit from SIM's structured latent spaces due to their encouraged smoothness, better disentanglement, and well-defined shapes.

6 Discussion

We presented Smooth InfoMax, a self-supervised representation learning approach that incorporates interpretability requirements into the design of the model. Our proposal demonstrates how β-VAE regularization can be integrated into GIM's contrastive learning framework at various depths in the network. As such, SIM enjoys GIM's computational advantages—such as decoder-less training, large-scale distributed training for architectures that would otherwise not fit in memory, and reduced vanishing gradients—while also preserving the well-structured latent space properties of β-VAEs across layers. Remarkably, this is achieved without significantly compromising performance. SIM enables more effective post-hoc interpretability, bringing us closer to understanding the internal workings of these neural networks.

Limitations and Future Work Although the latent space properties of β-VAEs improve post-hoc interpretability, the overall success still depends on the faithfulness of the generated explanations and the clarity of the information encoded in the representations. Additionally, SIM shows a small performance gap relative to its baseline, suggesting that integrating recent advances in disentanglement could be beneficial. While our evaluation focuses on sequential speech data (an XAI domain less exhaustively explored than vision), SIM's InfoNCE-based architecture is easily adaptable to other modalities, such as vision and natural language. Finally, SIM could be valuable beyond GIM as its probabilistic architecture and regularization can be integrated into other frameworks too, including end-to-end NNs as shown in this toy example.

Acknowledgments. This research was funded by the Department of Computer Science at the University of Antwerp and by the Vrije Universiteit Brussel.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

Bibliography

[1] Bau, D., Zhou, B., Khosla, A., Oliva, A., Torralba, A.: Network Dissection: Quantifying Interpretability of Deep Visual Representations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6541–6549 (2017)

[2] Bengio, Y.: Learning deep architectures for AI. Found. Trends Mach. Learn. **2**(1), 1–127 (2009)

[3] Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence **35**(8), 1798–1828 (2013)

[4] Bhati, S., Villalba, J., Zelasko, P., Moro-Velázquez, L., Dehak, N.: Segmental contrastive predictive coding for unsupervised word segmentation. In: Hermansky, H., Cernocký, H., Burget, L., Lamel, L., Scharenborg, O., Motlícek, P. (eds.) 22nd Annual Conference of the International Speech Communication Association, Interspeech 2021, Brno, Czechia, August 30 - September 3, 2021. pp. 366–370. ISCA (2021)

[5] Burgess, C.P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., Lerchner, A.: Understanding disentangling in β-vae. arXiv preprint arXiv:1804.03599 (2018)

[6] Chen, R.T., Li, X., Grosse, R.B., Duvenaud, D.K.: Isolating sources of disentanglement in variational autoencoders. Advances in neural information processing systems **31** (2018)

[7] Doersch, C.: Tutorial on variational autoencoders. arXiv preprint arXiv:1606.05908 (2016)

[8] Erhan, D., Bengio, Y., Courville, A., Vincent, P.: Visualizing higher-layer features of a deep network. University of Montreal **1341**(3), 1 (2009)

[9] Ge, Y., Xu, Z., Xiao, Y., Xin, G., Pang, Y., Itti, L.: Encouraging disentangled and convex representation with controllable interpolation regularization. In: IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2023, Waikoloa, HI, USA, January 2-7, 2023. pp. 4750–4758. IEEE (2023)

[10] Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: Gordon, G.J., Dunson, D.B., Dudík, M. (eds.) Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011. JMLR Proceedings, vol. 15, pp. 315–323. JMLR.org (2011)

[11] Hénaff, O.J.: Data-efficient image recognition with contrastive predictive coding. In: Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event. Proceedings of Machine Learning Research, vol. 119, pp. 4182–4192. PMLR (2020)

[12] Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A.: Beta-VAE: Learning Basic Visual Concepts

with a Constrained Variational Framework. In: International Conference on Learning Representations (Jul 2022)
[13] Hochreiter, S.: The vanishing gradient problem during learning recurrent neural nets and problem solutions. Int. J. Uncertain. Fuzziness Knowl. Based Syst. **6**(2), 107–116 (1998)
[14] Hsu, W., Zhang, Y., Glass, J.R.: Unsupervised learning of disentangled and interpretable representations from sequential data. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA. pp. 1878–1889 (2017)
[15] Kim, H., Mnih, A.: Disentangling by factorising. In: International conference on machine learning. pp. 2649–2658. PMLR (2018)
[16] Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: Bengio, Y., LeCun, Y. (eds.) 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings (2014)
[17] Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., Bachem, O.: Challenging common assumptions in the unsupervised learning of disentangled representations. In: international conference on machine learning. pp. 4114–4124. PMLR (2019)
[18] Löwe, S., O'Connor, P., Veeling, B.S.: Putting an end to end-to-end: Gradient-isolated learning of representations. In: Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada. pp. 3033–3045 (2019)
[19] Lu, M.Y., Chen, R.J., Wang, J., Dillon, D., Mahmood, F.: Semi-supervised histology classification using deep multiple instance learning and contrastive predictive coding. arXiv preprint arXiv:1910.10825 (2019)
[20] Mahendran, A., Vedaldi, A.: Visualizing deep convolutional neural networks using natural pre-images. Int. J. Comput. Vis. **120**(3), 233–255 (2016)
[21] Mordvintsev, A., Olah, C., Tyka, M.: Inceptionism: Going deeper into neural networks (2015), `https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html`
[22] Moshe, O., Fidel, G., Bitton, R., Shabtai, A.: Improving interpretability via regularization of neural activation sensitivity. Mach. Learn. **113**(9), 6165–6196 (2024)
[23] Nguyen, A.M., Dosovitskiy, A., Yosinski, J., Brox, T., Clune, J.: Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In: Lee, D.D., Sugiyama, M., von Luxburg, U., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain. pp. 3387–3395 (2016)
[24] Nhem, T., Denoodt, F., Weyn, M., Peeters, M., Oramas, J., Berkvens, R.: Label-efficient learning for radio frequency fingerprint identification. In: IEEE Wireless Communications and Networking Conference, WCNC 2025 (2025), to appear.

[25] Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
[26] Panayotov, V., Chen, G., Povey, D., Khudanpur, S.: Librispeech: an asr corpus based on public domain audio books. In: 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP). pp. 5206–5210. IEEE (2015)
[27] Sikka, H., Zhong, W., Yin, J., Pehlevant, C.: A Closer Look at Disentangling in β-VAE. In: 2019 53rd Asilomar Conference on Signals, Systems, and Computers. pp. 888–895 (Nov 2019)
[28] Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. In: Bengio, Y., LeCun, Y. (eds.) 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings (2014)
[29] Stacke, K., Lundström, C., Unger, J., Eilertsen, G.: Evaluation of contrastive predictive coding for histopathology applications. In: Alsentzer, E., McDermott, M.B.A., Falck, F., Sarkar, S.K., Roy, S., Hyland, S.L. (eds.) Proceedings of the Machine Learning for Health NeurIPS Workshop. Proceedings of Machine Learning Research, vol. 136, pp. 328–340. PMLR (11 Dec 2020)
[30] Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th international conference on Machine learning. pp. 1096–1103 (2008)
[31] Wiskott, L., Sejnowski, T.J.: Slow feature analysis: Unsupervised learning of invariances. Neural Comput. **14**(4), 715–770 (2002)
[32] Wu, M., Parbhoo, S., Hughes, M.C., Roth, V., Doshi-Velez, F.: Optimizing for Interpretability in Deep Neural Networks with Tree Regularization. Journal of Artificial Intelligence Research **72**, 1–37 (Sep 2021)
[33] Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D.J., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I. Lecture Notes in Computer Science, vol. 8689, pp. 818–833. Springer (2014)