VRIJE
UNIVERSITEIT
BRUSSEL

Master thesis submitted in partial fulfilment of the requirements for the degree of
Master of Science In de Ingenieurswetenschappen: Computerwetenschappen

VARIATIONAL GREEDY INFOMAX

Towards independent and interpretable representations

Fabian Denoodt

2022-2023

Promotor(s): Prof. Dr. Bart de Boer
Science and Bio-Engineering Sciences

VRIJE
UNIVERSITEIT
BRUSSEL

Proefschrift ingediend met het oog op het behalen van de graad van Master of Science In de Ingenieurswetenschappen: Computerwetenschappen

[DUTCH] VARIATIONAL GREEDY INFOMAX

[Dutch] Towards independent and interpretable representations

Fabian Denoodt

2022-2023

Promotor(s): Prof. Dr. Bart de Boer

Wetenschappen en Bio-ingenieurswetenschappen

Contents

Chapter 1

Relation to existing work

In recent years CPC is shown to be a successful self-supervised learning approach in a wide range of domains [1, 2, 3, 4, 5, 6]. Löwe et al. contribute to this work by showing that modules can be trained greedily, each with their own instance of the InfoNCE loss, allowing for training modules in parallel on multiple devices, or sequentially, offering a solution for hardware constrained devices. In addition, Wang [7] show that GIM can be applied to learn good speech representations from the speech dataset discussed in section **??**.

However, the underlying representations obtained from optimising the InfoNCE loss have not yet been studied in great detail. In this thesis we shed light onto this, through the introduction of a constraint to the latent space, causing space, which can be better understood and analysed. We have studied the latent representations obtained from maximising the InfoNCE objective. We achieved this through the introduction of V-GIM, a self-supervised representation learning approach with the same InfoNCE objective, but with an additional constraint to the latent space resulting in better interpretable representations. Such that a decoder could be trained and predict meaningful ...

1.1 Explainable ANNs

This is a vastly different approach from existing techniques in explainable AI. X et al. group XAI techniques in x cateogories In the field of Explainable AI multiple paradigms exist, ranging from activation heatmaps ...

These techniques give insights in visual domain, but lack in other domains such as speech domain where heatmaps be harder to gain insights from.

— Explainable ANNs: - diff paradigms, by looking at heath maps and lr etc - our work is in fact new paradigm, by adding constraints to the optimisation metric, resulting in better understandable latent representations - Explainable deep learning methods survey: [8] (attribution and non attribution, zie mijn draft.dox) –¿ probeert contribution van elke feature te linken. dat zijn technieken die werken voor foto's of feature vectors, maar voor puur sequential audio is moeilijker bruikbaar.

Maybe exists other techniques that change the ANN resulting in better explainable. (eg pruning?) - methodology to remove features that do not contribute to accuracy. (feature selection) with interpretability motivations. [9]

Explainable representation learning in speech data

Explainable representation learning - Representation learning overview: eventueel enkel opnoemen? wasserstijn gans etc?

- InfoGAN: chenInfoGANInterpretableRepresentation2016, guoDeepMultimodalRepresentation2019 This idea of constraining the latent space to become more interpretable was also done in InfoGAN. which learns disentangled and interpretable representations. achieved by maximising mutual between a small subset of the latent variables and the observation [10].

Variational learning (alternative priors): Taking inspiration from VAEs, we constrain V-GIM's latent space by minimising the KL-divergence to a standard normal Gaussian. In recent years, multiple extensions to VAEs have been introduced which use a different prior. While the most common posterior is a factorised gaussian for easy mathematical derivation, Kingma et al [11] show that alternative full Gaussian is also possible. ref Kinmga thesis? "Variational learning (alternative priors): cite:nalisnickApproximateInferenceDeep". "a factorized Gaussian, thereby imposing strong constraints on the posterior form and its ability to represent the true posterior, which is often multimodal" Full-covariance Gaussian posterior.

Alternative priors - [12]: use a mixture of Gaussians and results in less useless latent dimensions. - Gaussian mixture model: [13] - Gaussian mixture: [14] - Hierarchical VAE: [15]

1.2 Representation learning: explainable

1.3 Variational learning

There already were a few papers with variational contrastive predictive coding

1.4 Links I should investigate

- S3VAE: Self-Supervised Sequential VAE for Representation Disentanglement and Data Generation https://arxiv.org/abs/2005.11437

- Implementation of Sequential VAE https://github.com/ermongroup/Sequential-Variational-Autoencoder

- Contrastively Disentangled Sequential Variational Autoencoder https://proceedings.neurips.cc/paper/2021/f
Paper.pdf

- Sequential Variational Autoencoders for Collaborative Filtering https://arxiv.org/pdf/1811.09975.pdf

- !! Variational noise contrastive estimation: https://arxiv.org/abs/1810.08010

Chapter 2

Discussion

– decaying learing rate: we train using decaying lr, because models must first learn distributions and goes too slow if lr is too small. and a learning rate scheduler ExponentialLR decay rate 0.995

— batch norm: - sindy didn't have issues of batch norm, but believe this is because each module consisted of a single layer, ours contain a number of layers. potentially: outputs from first module change too fast for second module to catch up.

while GIM argues to resolve memory constraints, not entirely true. In fact we even countered the opposite as containing multiple neural networks, each with their own personal loss function (the loss function is based on fk which contains parameters that must be learned), and thus for early layers where the sequence is still long, a lot of memory is required. We went for a compromise on GIM by splitting up the architecture in merely two modules, significantly reducing the memory constraints.

— The second module in GIM clearly doesn't have as much effect. This can be explained because there may not be as much common information anymore between the patches. There may be a source that says that cpc learns low level features, but the second module is supposed to learn more high level features, which cpc may have trouble with? —

Future work: - Related work in VAE shows that gradually increasing regularisation term, results in better disentengledment, while avoiding posterior collapse. could have a kldweight scheduler.

- not constrained solely to InfoNCE loss, the GIM architecture could work for other losses too that allow for greedy optimisation.

- I didn't add an autoregressor as i didn't find a performance benefit. Potentially, with larger architecture could further improve performance.

—- Towards production setting: encodings are thus optimised to be close the standard normal. When in a production environment and new data is given, could in fact have an idea of how well generalisation to the production data: eg via anomaly detection if encodings are too far away from center. = gives automated way of verifying generalisation.

can then maybe see to which data that doesn't generalise well via outliers.

—-

future work: - disentanglement should do more investations

— GIM: Modular training could incrementally increase numb of modules and observe performance increase for downstream tasks. based on this, could find smallest gim architecture depth which satisfies required accuracies.

—- interpretability: most dimensions sensitive around 75 to 150 hz. this is as expected as the adult man speaks around 80 to 180 hz.

— interpretability is only as good as the decoder. if a shitty decoder doesn't construct well, doesn't necessarily give correct conclusions about V-GIM.

- Explainability of latents is dependent on the performance of the decoder.
- Intermediate loss function with kld resulted in similar behaviour as batch normalisation. Resulting in faster convergence than without kld.
- We observed no quality loss in the learned representations. Data was equally easily separable.

Bibliography

[1] K. Stacke, C. Lundström, J. Unger, and G. Eilertsen, "Evaluation of Contrastive Predictive Coding for Histopathology Applications," in *Proceedings of the Machine Learning for Health NeurIPS Workshop*. PMLR, pp. 328–340. [Online]. Available: https://proceedings.mlr.press/v136/stacke20a.html

[2] p. u. family=Haan, given=Puck and S. Löwe. Contrastive Predictive Coding for Anomaly Detection. [Online]. Available: http://arxiv.org/abs/2107.07820

[3] M. Y. Lu, R. J. Chen, J. Wang, D. Dillon, and F. Mahmood. Semi-Supervised Histology Classification using Deep Multiple Instance Learning and Contrastive Predictive Coding. [Online]. Available: http://arxiv.org/abs/1910.10825

[4] S. Bhati, J. Villalba, P. Żelasko, L. Moro-Velazquez, and N. Dehak. Segmental Contrastive Predictive Coding for Unsupervised Word Segmentation. [Online]. Available: http://arxiv.org/abs/2106.02170

[5] S. Deldari, D. V. Smith, H. Xue, and F. D. Salim, "Time Series Change Point Detection with Self-Supervised Contrastive Predictive Coding," in *Proceedings of the Web Conference 2021*, ser. WWW '21. Association for Computing Machinery, pp. 3124–3135. [Online]. Available: https://dl.acm.org/doi/10.1145/3442381.3449903

[6] O. Henaff, "Data-Efficient Image Recognition with Contrastive Predictive Coding," in *Proceedings of the 37th International Conference on Machine Learning*. PMLR, pp. 4182–4192. [Online]. Available: https://proceedings.mlr.press/v119/henaff20a.html

[7] Meihan Wang, "Speech representation learning without backpropagation."

[8] X. Bai, X. Wang, X. Liu, Q. Liu, J. Song, N. Sebe, and B. Kim, "Explainable deep learning for efficient and robust pattern recognition: A survey of recent developments," vol. 120, p. 108102. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320321002892

[9] L. W. Glorfeld, "A methodology for simplification and interpretation of backpropagation-based neural network models," vol. 10, no. 1, pp. 37–54. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/0957417495000321

[10] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. [Online]. Available: http://arxiv.org/abs/1606.03657

[11] D. P. Kingma and M. Welling, "An Introduction to Variational Autoencoders," vol. 12, no. 4, pp. 307–392. [Online]. Available: http://arxiv.org/abs/1906.02691

[12] J. Tomczak and M. Welling, "VAE with a VampPrior," in *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 1214–1223. [Online]. Available: https://proceedings.mlr.press/v84/tomczak18a.html

[13] C. Guo, J. Zhou, H. Chen, N. Ying, J. Zhang, and D. Zhou, "Variational Autoencoder With Optimizing Gaussian Mixture Model Priors," vol. 8, pp. 43 992–44 005.

[14] E. Nalisnick, L. Hertel, and P. Smyth, "Approximate Inference for Deep Latent Gaussian Mixtures."

[15] A. Vahdat and J. Kautz, "NVAE: A Deep Hierarchical Variational Autoencoder," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., pp. 19 667–19 679. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/e3b21256183cf7c2c7a66be163579d37-Abstract.html

Appendices

Appendix A

Syllable classification through histogram segmentation

probabily bs: During inference, (in this context obtaining the latent representations for our input signals), depending on the length of the input signal, the length of the output latent representation will differ. If we wish to look at how separable latent representations are for syllables, the length can be variable. Some input sounds could be 6,600 samples, while others 8,800 samples. We therefore pad the syllables with zeroes in front and end of the signal, to obtain fixed length of equal to that of the longest syllable; 8,800 samples.

Training happens on longer data samples, and every **X** epochs t-SNE visualisations are made to observe evolutional of dis-entanglement.

Since the recordings are very consistent in loudness and are noise free, we can split up the files per syllable, obtaining three files per original sound (one for each syllable).

A sliding window is used of size 0.02 seconds. With a sample rate of 22050, this corresponds to roughly 500 samples per window. The maximum is computed for each window. Speech signals can then be split up when a severe dip happens in the signal. Regions where the amplitude is greater than 0.2 are considered **klinkers**, the regions with with lower values are considered **medeklinkers**. Apart from a few edge cases, this technique worked well enough for this purpose. In those cases, the splitting points closest to the one-third and two-third splitting points were considered.

ERR: OUDE AFBEELDINGEN ZIJN WEG

note: we do need a hard threshold which is based on the signal's intentisity level. One could consider the alternative approach of looking at the gradient at each point and selecting the points with largest negative gradient. This will work in many cases, however, not for temporal envelops which gradually move towards zero, s.t: A.1. Instead we use a dynamic threshold. This threshold is computed by creating transforming the signal into bins of 90'th percentile, creating a histogram of the single signal and applying otsu's image segmentation algorithm to obtain the threshold of that single audio sample. We also tried directly applying otsu to the moving average and maximum of the bins. This either gave a threshold that was too small or too large. the 90th percent resulted in an acceptable compromise.

Example where the explained strategy does not work:

Reference images for in the text:

audio padded to maximum length. (added zeros in front and back)

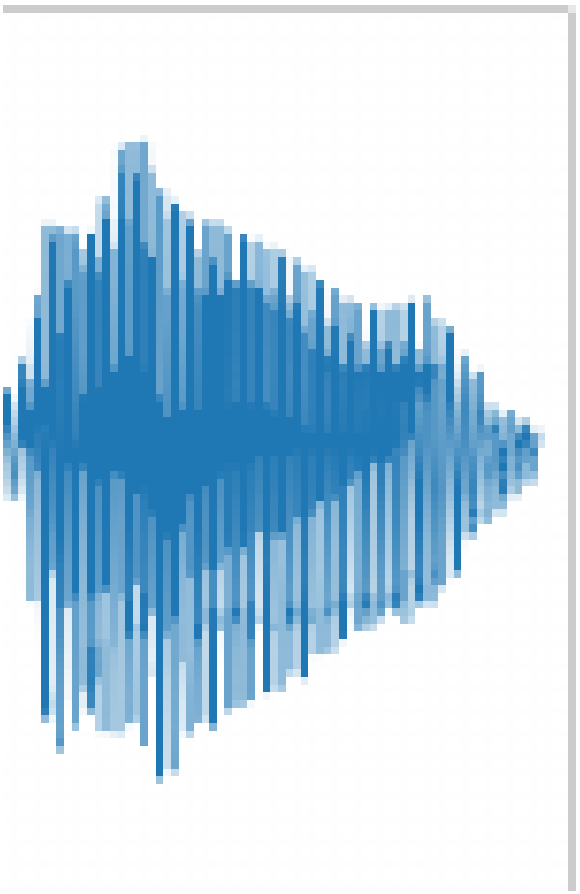

Figure A.1

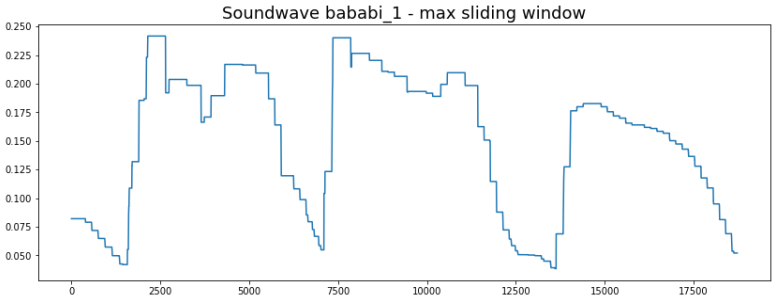

Figure A.2

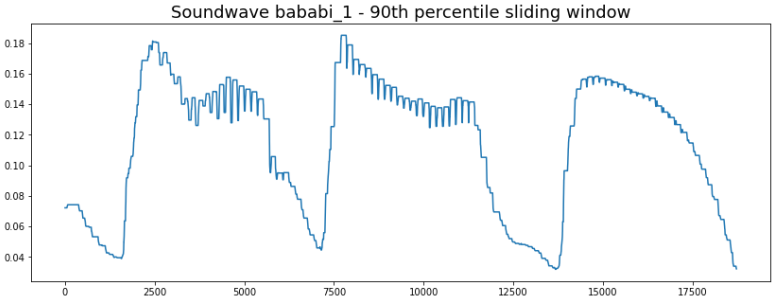

Figure A.3

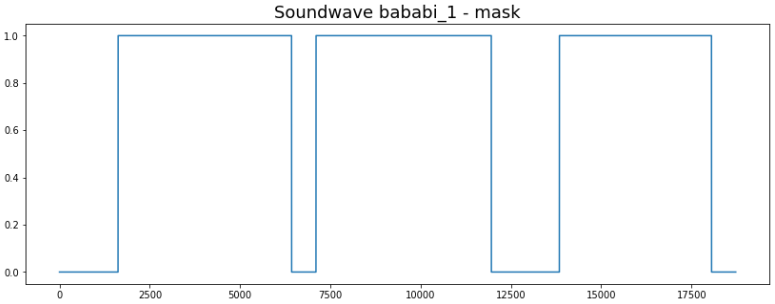

Figure A.4

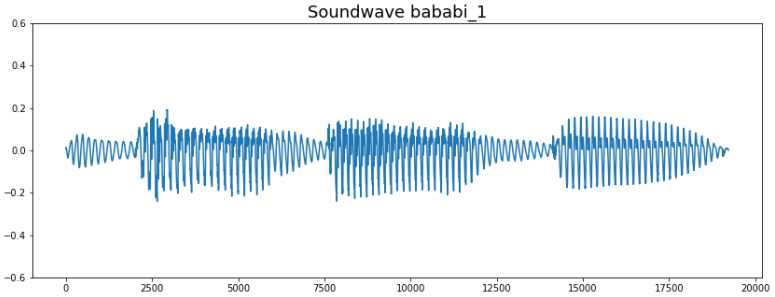

Figure A.5

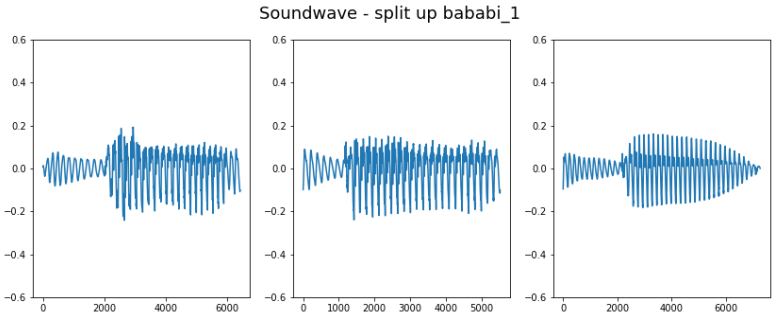

Figure A.6

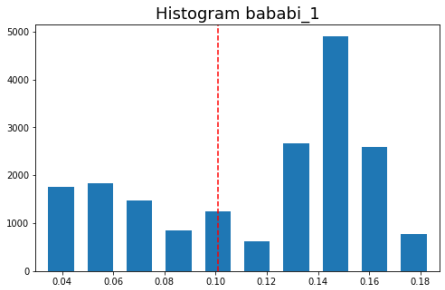
Histogram bababi_1
5000
4000
3000
2000
1000
0
0.04
0.06
0.08
0.10
0.12
0.14
0.16
0.18

Figure A.7

Appendix B

Some more appendix

B.1 GIM: Activations visualisations

thought for later: its actually weird i was able to play enc as audio as enc is 512 x something so huh? that means that a lot of info is already in first channel? what do other 511 channels then contain? """ Observations: First layer decoded still contains the same sound, but with some added noise (could be because decoder hasn't trained very). However, the encoded first layer, still contains the exact sound as the original sound. It is however downsampled a lot -¿ from 16khz to 3khz """ thought for later: its actually weird i was able to play enc as audio as enc is 512 x something so huh? that means that a lot of info is already in first channel? what do other 511 channels then contain?

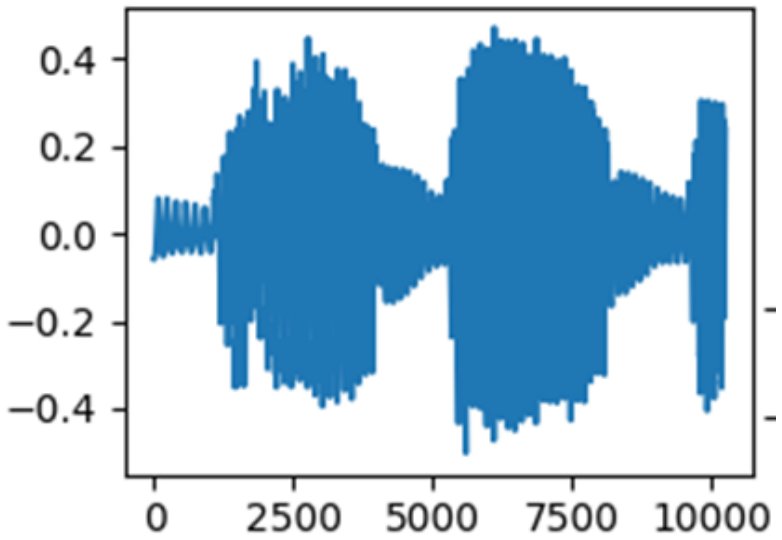

Figure B.1: "BA-BA-BA" time domain

No batch normalisation, so although channels appear to have larger activations than other channels, size of activation does not really say anything about information. eg activations 0.01 could still contain more information than 3.0 activation.

Figure B.2: Activations of the sound "BA-BA-BA" through GIM

Since the activations from convolutional neural networks, the order is still maintained. Hence, can align activations with original signal.

Observations in latent representations:

Layer 1: The activations of the first decoder still contain a lot of similarity with the original signal, in terms of structure. There is a lot of redundant data within the representation. Eg: the one channel could be replied

Layer 2

Layer 3:

Layer 4: Still notices multiple channels which have high activations when signal is has high amplitudes and small activations when amplitude is low.

Also activations which are high when volume is low. –¿ indicates that certain kernel weights are sensitive for **"klinkers"** and other kernels for **medeklinkers**. see B.3.

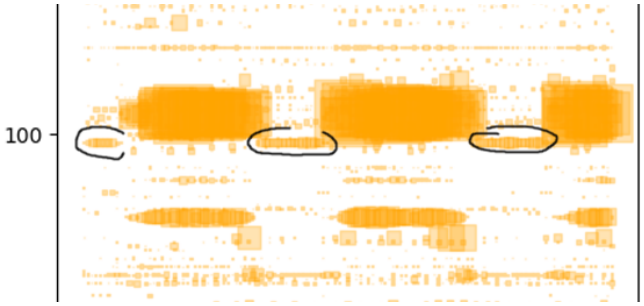

Figure B.3: zoomed in

Observe that activations happen in clusters/sequences. So it is usually a patch of signal samples that cause high activations. This could for instance indicate that both kernels are sensitive for the **medeklinker** "b", but sensitive for different features. eg the letter B has spoken sound "buh". so maybe one is sensitive for "b" and other for "uh".

Figure B.4 also nicely shows how different channels have clusters of activations at slightly different times.

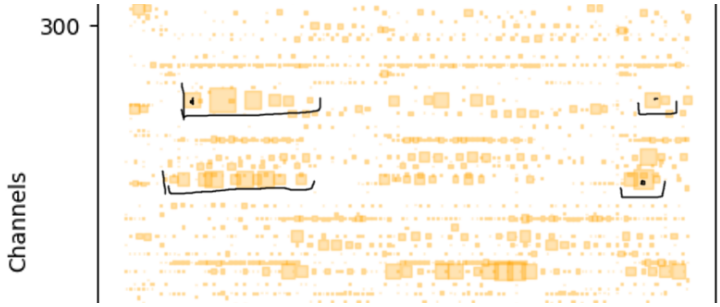

Figure B.4: Zoomed in

Appendix C

Interpolation

Graphs related to the interpolation experiment are shown in figures C.1, C.2, C.3. We categorise the dimensions in V-GIM in three sections, the graphs present an example of each category.

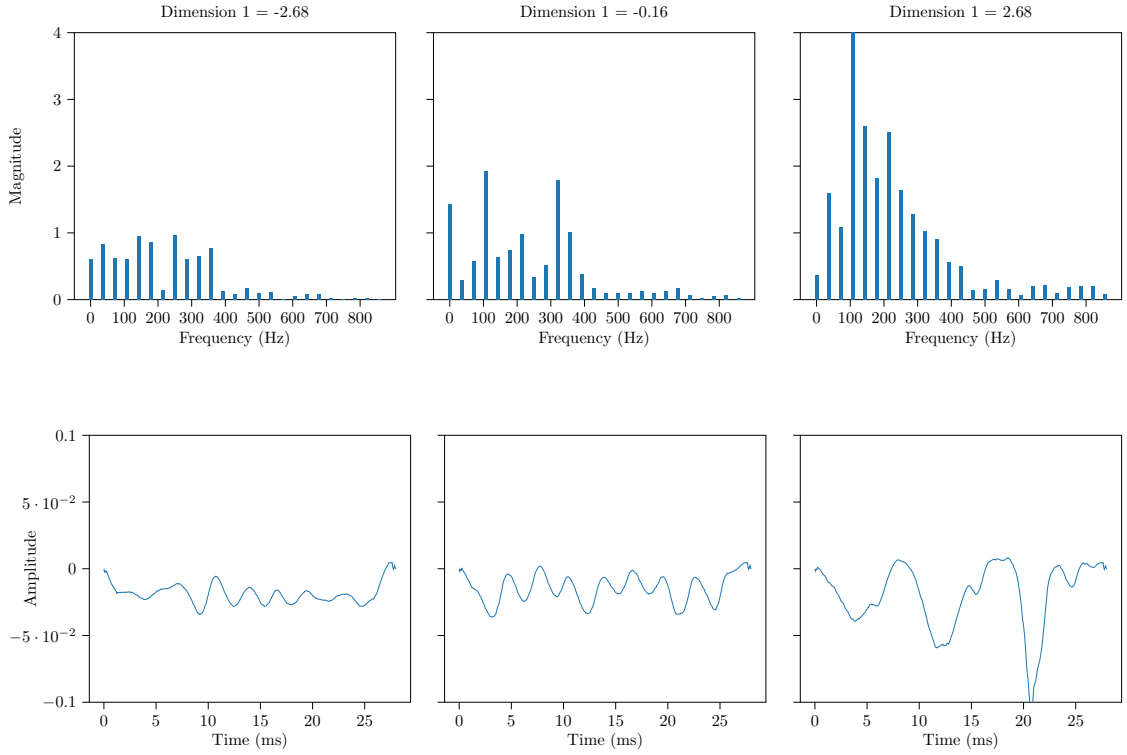

Figure C.1: The first dimension is being modified, while other dimensions are fixed at 0. The dimension has influence on the 150Hz frequency band, but also neighbouring ranges.

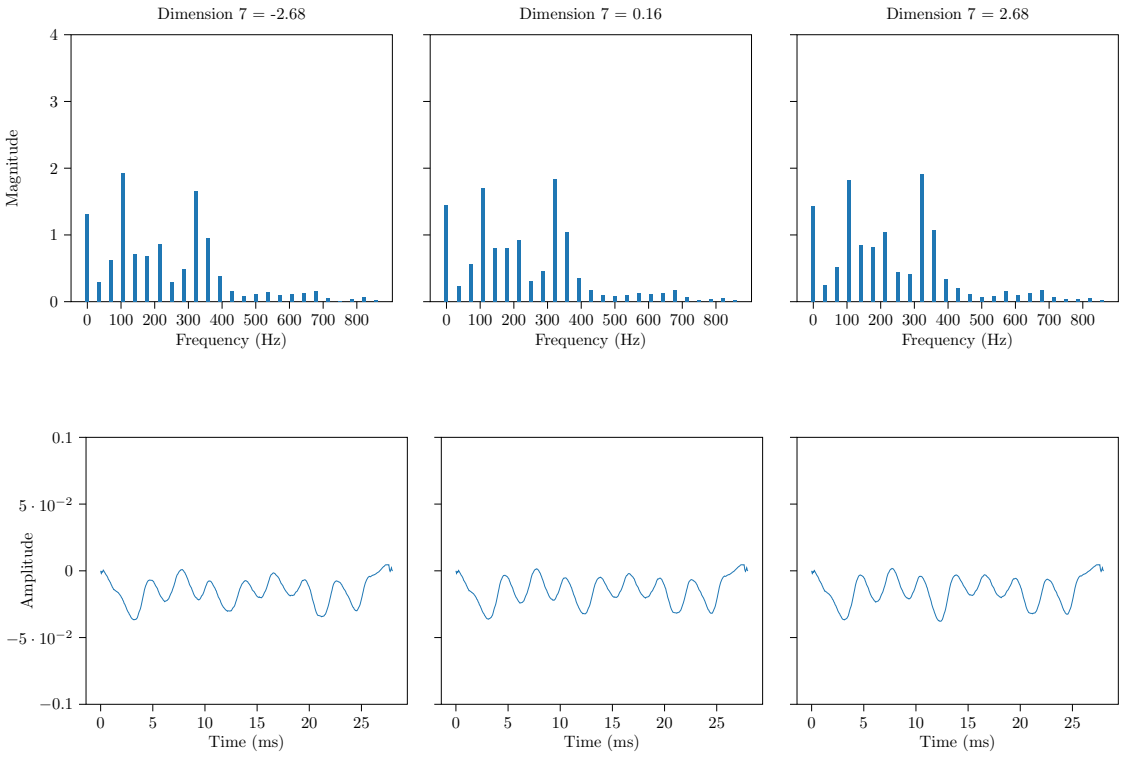

Figure C.2: The seventh dimension is being modified, while other dimensions are fixed at 0. We observe no significant differences when adjusting, indicating that the dimension may not capture any information at all.

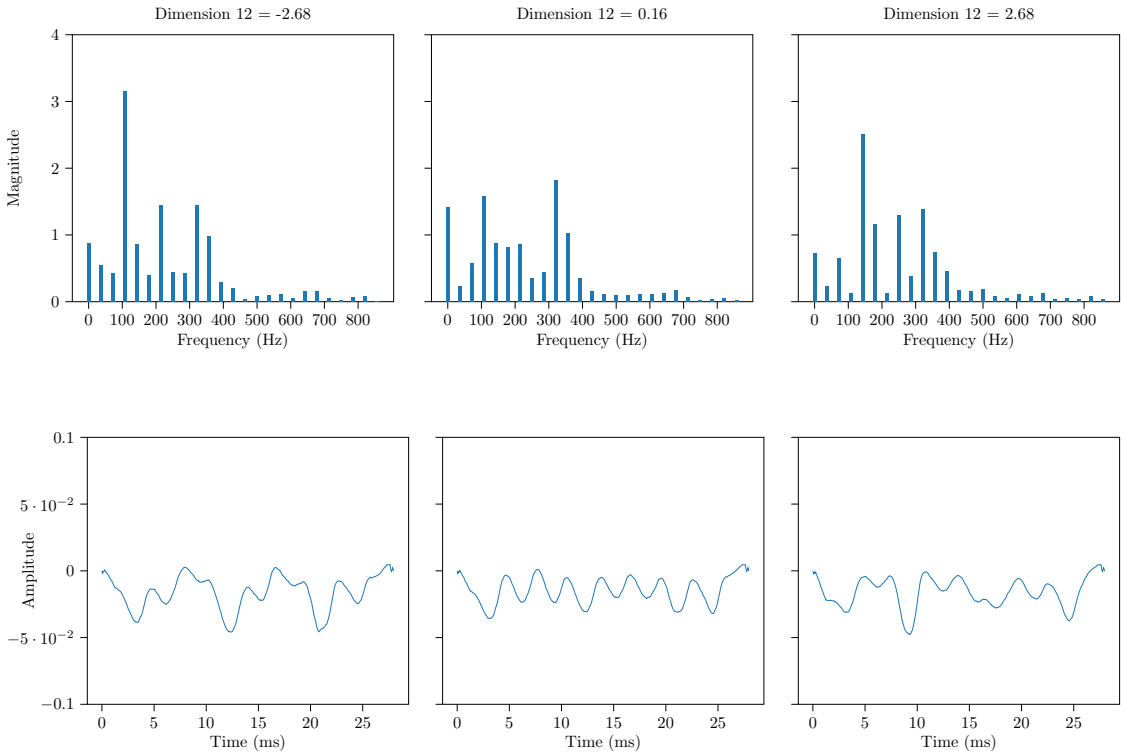

Figure C.3: The twelfth dimension is being modified, while other dimensions are fixed at 0. Negative values cause a large spike around 100Hz while positive values fully remove the 100Hz and influences the 150Hz range instead.