Vibrotactile Signal Compression using Perceptually Trained Autoencoders

Lars Nockenberg[1,2][0000−0001−5171−3850] and
Eckehard Steinbach[1,2][0000−0001−8853−2703]

[1] Technical University of Munich, School of Computation, Information and Technology, Department of Computer Engineering, Chair of Media Technology and Munich Institute of Robotics and Machine Intelligence (MIRMI)
[2] Centre for Tactile Internet with Human-in-the-Loop (CeTI), Technische Universität Dresden, Germany
{lars.nockenberg, eckehard.steinbach}@tum.de

Abstract. Haptic feedback is becoming a crucial element for enhancing immersion in various media applications. To enrich this feedback, high-quality haptic content, an appropriate playback device, and efficient codecs for transmission are essential. This paper introduces a novel vibrotactile codec that employs an autoencoder architecture integrated with Convolutional Neural Networks (CNNs). It leverages a tailored perceptual model with a band structure derived from the audio domain, optimizing the perceived quality of the encoded signals during training. Additionally, we have developed and assessed multiple perceptual training losses to further enhance the performance of our codec.

Keywords: haptics · compression · deep learning · haptic perception.

1 Introduction

Haptic feedback increases the immersion within virtual reality by introducing the sense of touch to experiences previously dominated by visual and auditory stimuli. In teleoperation scenarios, haptic feedback enhances the interaction quality by presenting the operator the crucial sensory cues of the interaction between the remote robot and physical objects. For instance, a study cited as [13] showcased a multitouch vibrotactile display, utilizing a 2D array of piezoelectric actuators. Meanwhile, Shanmugam et al. [5] highlighted the diverse array of haptic gloves developed for various applications, notably in virtual reality and teleoperation, demonstrating the broadening scope and potential of haptic technologies. As the utilization of haptic feedback in new applications continues to rise, high-quality compression techniques become increasingly crucial to reduce network load.

Vibrotactile compression aims at encoding single-channel or multi-channel vibrotactile data into an efficient representation. Especially with multi-channel devices, the relatively low data rate of vibrotactile signals can add up quickly. One of the essential principles of compression is to remove redundant or expendable data. In this context, expendable means that components of the signal are not perceived by a user and can therefore be omitted.

Understanding the nature of human perception and incorporating the findings into the compression architecture is vital for good performance. One of the important aspects in this context is how humans rate the similarity between stimuli, which Richardson et al. [15] aimed to predict from real interactions with surfaces. The work in [16] presents a model for extracting perceptual features from multimodal data obtained from surface interaction. Metzger and Toscani [7] used an autoencoder to encode vibrotactile signals into a compact representation and showed that the properties of the latent space predict human tactile material classification and resemble the structure of a perceptual tactile space. This shows that using autoencoders for compressing vibrotactile signals is an appropriate option when preserving perceptual properties of haptic textures is a primary goal.

Several vibrotactile codecs have already been proposed, with two recent conventional codecs being the Vibrotactile Compression with Perceptual Wavelet Quantization (VC-PWQ) [11] for single-channel and the Multi-Channel Vibrotactile Codec using Hierarchical Perceptual Clustering (MVibCode) [8] for multi-channel signals. The VC-PWQ uses a Discrete Wavelet Transform for decorrelation and has a perceptual model that analyzes the perceptual properties of a vibrotactile signal to aid the iterative distribution of bits for quantization. The separation into bands done by the transform makes it possible to optimize the quantization depth for different frequency components of the signal independently. The MVibCode uses this as a basis and groups the channels of a signal into clusters to encode them jointly using differential coding.

Deep learning based codecs are emerging as well, making use of the ability of neural networks to compress data into a compact latent representation. Zhao et al. [20] introduced a GRU-based predictive codec. It uses nonlinear quantization and Huffman coding to code the residual signals. Its strengths are low latency and high quality. In [21], a transformer architecture is used as an autoencoder. It uses a nonlinear quantizer and Huffman coding as well. Li et al. developed an autoencoding vibrotactile codec using a CNN.

All the mentioned deep learning based compression approaches do not utilize a perceptual model or metric for training or coding, which might hinder its performance. Contrary to [16] for example, we refer to a perceptual model as a tool to rate the perceptual importance of frequency components in a vibrotactile signal, only relying on this single modality. This is to suit our specific needs for the compression of vibrotactile signals. Knowing which parts of a signal are important for high perceptual quality is essential for good compression performance, even if that means lower objective quality of the output. Inspired by the recent success of deep learning-based approaches in compression, in this paper we present a new vibrotactile autoencoder-based codec that is trained on a perceptual loss. The perceptual loss is applied in the frequency domain and expands on the perceptual model that was developed in [11] to fit the needs of a training loss. We also investigated the concept of critical bands in this context, which are already utilized in audio codecs similarly. Critical bands were intro-

duced in [22], further modeling the perceptual effects that occur from masking in the frequency domain. Summarizing, our contributions are:

- Vibrotactile codec using a Resnet-based autoencoder derived from [4]
- Perform training using a perceptual loss with a band structure
- Comparison of training with and without perceptual loss, as well as different loss function approaches

2 Perceptual Vibrotactile Autoencoder

2.1 Architecture Overview

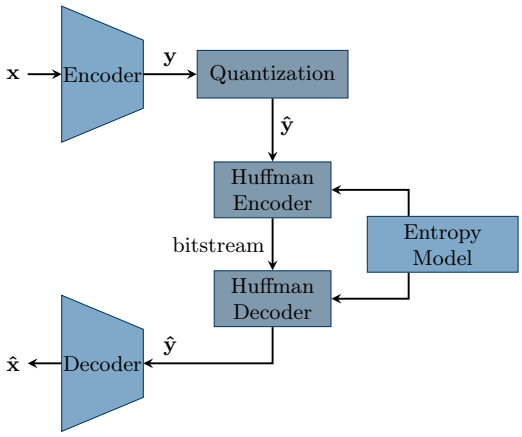
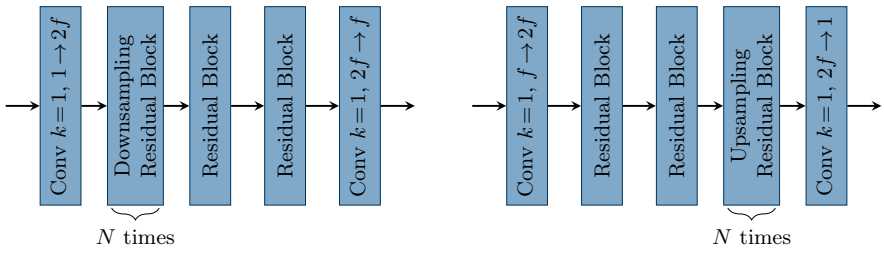


Fig. 1: Codec Architecture

The codec follows the basic structure of an autoencoder which is trained end-to-end. Its components are depicted in Fig. 1, which are the encoder and decoder network, a quantizer, and an entropy encoder and decoder, which are steered by an entropy model. Different from related approaches in image/video compression, the proposed codec does not use a hyperprior network to aid the entropy model. Presumably due to the much smaller block size compared to image compression, adding it in was not beneficial for the overall performance, while it would add to the computation time. The codec is block-based with a block length bl of 512 samples. For lower latency, this size can be reduced. The encoder and decoder consist of a CNN with residual connections, inspired by [4]. The advantage of this structure is, that deeper structures with more downsampling layers are possible without having to sacrifice performance. This is important since we utilize the encoder to reduce the number of latent samples for higher compression, so higher compression requires a deeper network. The structure of the encoder is shown in Fig. 2a. It has N residual blocks that each apply downsampling by factor 2 and

2 additional residual blocks at the end. N was varied from 1 to 4 in our tests. The latent size was varied from $(f = 1) \times bl/(2^N)$ to $(f = 2^N) \times bl/(2^N)$, such that with a relatively high N, by increasing feature size f still a low compression ratio could be achieved if desired. To achieve different feature sizes f with this architecture, a convolutional layer with kernel size 1 is added to the front that brings the feature size to $2f$ and another one at the back of the network that reduces it to the target f. The decoder is a mirrored version of this with the residual blocks in front and the upsampling ones afterwards.

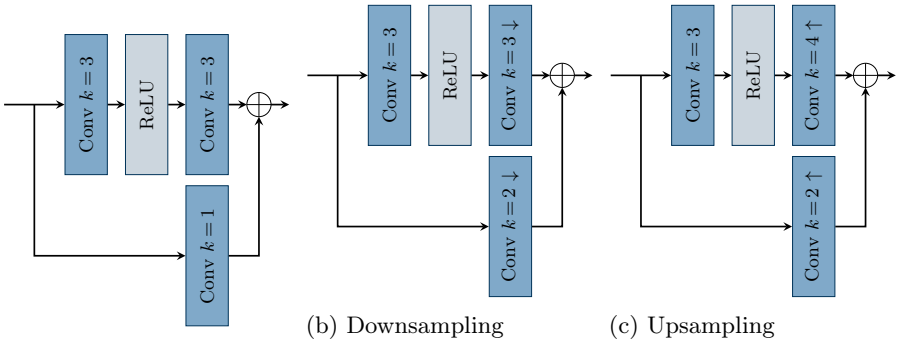


(a) Encoder (b) Decoder

Fig. 2: Encoder and decoder networks. The Conv blocks are convolutional layers. Their parameters are specified as "kernel size, input features → output features".

The residual blocks consist of two convolutional layers with a ReLU layer in between, as can be seen in Fig. 3a. The skip connection is added after applying a convolutional layer with kernel size $k=1$ to it that allows the model to weight the skip connection and therefore scale the impact of the convolution and non-linearity. The residual block with downsampling, displayed in Fig. 3b, has the same structure, just with the second convolutional layer having a stride of 2 for downsampling, also the skip connection goes through a convolutional layer with kernel size 2 and stride 2. So, the receptive field of the skip connections is kept to a minimum.

Uniform 7 bit quantization is applied after the encoder. The entropy model is based on [1]. It is a learned entropy model consisting of linear layers and nonlinearities designed in such a way that the network is forced to form a Cumulative Density Function while being able to derive the Probability Density Function from it. As entropy coding method, Huffman coding is applied. With end-to-end trained autoencoders, often Context-Adaptive Binary Arithmetic Coding (CABAC) is utilized. CABAC is capable of adapting to the input distribution at runtime, which is necessary if a hyperprior is used. Since we do not use a hyperprior in our codec, Huffman coding can be used without having to generate a new Huffman table for each block. Because of its low computational complexity, this makes Huffman coding the more efficient method to use.



(a) Residual Block (b) Downsampling Residual Block (c) Upsampling Residual Block

Fig. 3: Residual Block Structures

2.2 Training Loss

The training loss consists of an entropy loss and a distortion loss. For the latter, often the Mean Squared Error (MSE) is used, but we will investigate new options here, i.e. using a perceptual loss that quantifies the perceived loss of signal quality. This loss is derived from the perceptual model in [11], which was called the Psychohaptic Model (PM).

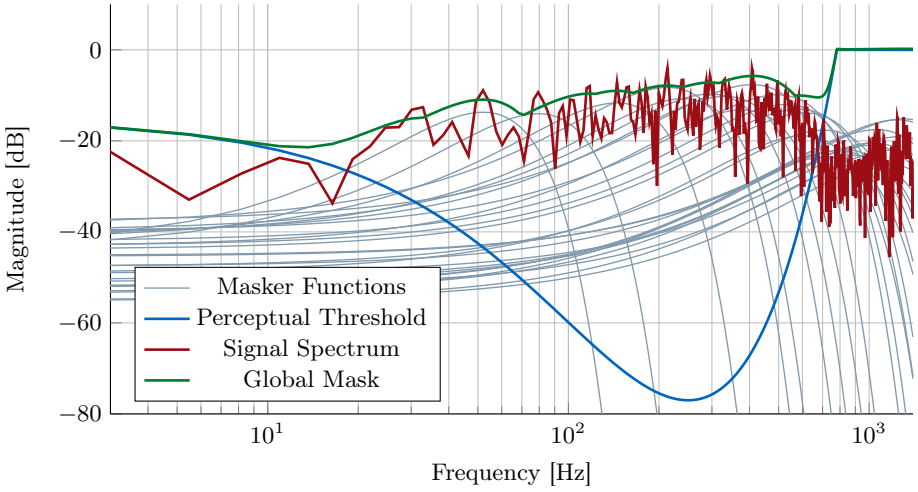


Fig. 4: Example output of the Psychohaptic Model. The masker functions correspond to the peaks found in the exemplary signal spectrum and their maximum is added to the signal-independent perceptual threshold to form the global mask.

The PM takes the absolute threshold of perception as well as masking into account. Its result comes in form of a Mask-to-Noise Ratio (MNR) calculated for individual frequency bands, indicating how high the level of noise is compared to the masking that occurs in that specific band. The model takes the input signal in the frequency-domain and analyzes it for frequency masking effects. We use the Discrete Cosine Transform as transform here. A masking threshold is formed by analyzing the spectrum of the signal to compress for peaks and generating a masker function for each peak. The maximum over all masks at each frequency forms the masking threshold. This is then combined with the threshold of perception by adding their spectra to form a global mask. The threshold of perception is a model for the frequency dependence of the human sense of touch. It represents how high the amplitude of a stimulus has to be at a certain frequency to be perceivable and was derived from real measurements in [9] and refined in [11]. In general, the sensitivity is lower for low and high frequencies and higher in the midrange. We use the threshold as defined in [11] and displayed in Fig. 4, where also exemplary masker functions and the resulting global mask are visualized.

Then, a Signal-to-Mask Ratio (SMR) is computed that quantifies how high the signal is above its corresponding mask spectrum:

$$SMR(b) = \frac{E_{signal}(b)}{E_{mask}(b)}, \tag{1}$$

where b is the band index, $E_{signal}(b)$ is the signal energy in band b and $E_{mask}(b)$ is the corresponding energy of the mask spectrum in that band. The separation into bands was originally used because of the wavelet transform used in the VC-PWQ. The SMR is a measure of how perceptible the individual frequency components of a signal are. From the SMR and the Signal-to-Noise Ratio (SNR), a Mask-to-Noise Ratio (MNR) can be calculated to judge the quality of the quantized representation of each band:

$$MNR(b) = SNR(b) - SMR(b), \tag{2}$$

The MNR can be interpreted as a measure of how perceptible the reconstruction noise in a given frequency band is. For more details on how the PM was designed, the interested reader is referred to [11].

To form a perceptual training metric from the SMR, two main approaches are considered. One is to weight the reconstruction noise using the SMR directly, which we will call the SMR-weighted loss. The other is to form bands and calculate the MNR for each of the bands. This we will refer to as the MNR-based loss. To obtain a single-valued loss, the average of all bands is calculated. This way, the optimizer will try to balance the MNR between bands to create an even perceived signal quality across all bands. The VC-PWQ uses a wavelet transformation, which naturally makes it necessary to split the psychohaptic analysis into bands. In this case, however, the band structure is not determined by the decorrelating transform and its design should be discussed.

In MPEG audio compression, the perceptual model is calculated in critical bands [12]. Critical bands are used in audio processing to model the masking phenomena and were introduced by Zwicker et al. [22]. They form a system that can be interpreted as a filterbank that separates a spectrum into bands, with each band fit to a selected center frequency with a bandwidth matching its masking properties. So, as pointed out in [22], critical bands do not have a fixed center frequency defined by the physical principles behind perception. Their bandwidth can be fixed to the -3 dB-points of the masker functions at the center frequency, as it is usually done for filters. Since the perceptual measurements are subject to error, making precise adjustments to these parameters does not necessarily make sense.

[6] and [2] showed that critical bands can also be applied to haptic perception. Picinali et al. compared audio and haptics in a two-tone discrimination task, finding similarities in the overall phenomena, but an overall lower sensitivity in haptics [14]. The masker functions developed in [11] have a bandwidth that increases logarithmically with frequency. Therefore, a model with critical bands should also have logarithmically increasing bandwidth. In [6] and [2], it has been shown that no masking occurs between the pacinian and non-pacinian channels. According to [2], the crossover frequency of the two channels is located at 40 Hz. This is a further indicator that the concept of critical bands is applicable for vibrotactile compression as well. With the VC-PWQ, each band is double in size compared to the next lower one besides the two lowest bands, which have the same size. Therefore, the logarithmic increase in bandwidth is already built in. With 2800 Hz sampling rate and a block length of 512 samples, one of the band boundaries is already close to 40 Hz, with 43.75 Hz. At a frequency resolution of roughly 2.73 Hz, this is only one frequency bin off, so the band separation there can already be interpreted as a good approximation. This also makes a direct comparison to the VC-PWQ easier, so we kept this band structure in our experiments.

In the following, the final training losses are derived. The MNR for the MNR-based loss is obtained from

$$MNR(b) = 10 \log_{10} \frac{E_{mask}(b)}{E_{noise}(b)}, \tag{3}$$

with b being the band index, $E_{mask}(b)$ and $E_{noise}(b)$ are the global mask and noise energy summed over the band b, respectively. This is an alternative way of calculating the MNR to Eq. 2, only requiring the global mask and noise spectra. It is now possible to try two different approaches for the global mask: with and without the masking threshold. This means that we can skip the addition of the masking threshold and only obtain the global mask from the threshold of perception, thus getting a mask independent from the signal to compress. To get a single-value loss L_{P1} for training, we take the average and adapt the value range:

$$L_{P1} = -\frac{1}{20}\sum_{b=1}^{B} MNR(b) - 22 \tag{4}$$

B is the total number of bands. For the SMR-weighted loss L_{P2}, the SMR in linear domain, here calculated by frequency bin, is taken to weight the MSE and form a new MSE_w. To keep the value range of the MSE, it is then again weighted by the sum of all SMR-coefficients:

$$L_{P2} = \sum_{i=0}^{bl-1} \frac{1}{SMR[i]} \cdot \sum_{i=0}^{bl-1} SMR[i] \cdot (\mathbf{x_{DCT}}[i] - \hat{\mathbf{x}}_{\mathbf{DCT}}[i])^2 \tag{5}$$

The rate loss is obtained through the learned entropy model. We calculate it using approximated discrete probabilities of the interval $[\hat{\mathbf{y}} - \frac{1}{2}\Delta, \hat{\mathbf{y}} + \frac{1}{2}\Delta]$ with Δ being the quantization interval. This ensures that the impact of the quantization bin size on the bitrate influences the optimization. The rate loss is calculated as:

$$R = \sum_{\hat{y}} -log(p_{\hat{\mathbf{y}}}(\hat{\mathbf{y}})) \tag{6}$$

$$p_{\hat{\mathbf{y}}}(\hat{\mathbf{y}}) = P\left(\hat{Y} \leq \hat{\mathbf{y}} + \frac{1}{2}\Delta\right) - P\left(\hat{Y} \leq \hat{\mathbf{y}} - \frac{1}{2}\Delta\right) \tag{7}$$

The final training loss with $L_{P1/2}$ as placeholder for both perceptual losses L_{P1} and L_{P2} is

$$L = \lambda_1 \cdot L_{P1/2} + (1 - \lambda_1) \cdot MSE + \lambda_2 R. \tag{8}$$

3 Results

3.1 Training

Training was carried out using the dataset recorded by Strese et al. [17]. They recorded data from a large variety of surfaces using the Texplorer2, a probe that can be slided over a surface and captures multiple modalities, including vibrotactile and audio signals.

[18] is a database recorded by sliding a stylus with an attached accelerometer over several surfaces. Both the type of the stylus' tip and the exploration speed was varied to obtain a larger database with a variety of signals. This dataset was already used to evaluate the VC-PWQ and the codec from [20] for example, and using the above-mentioned training dataset enabled us to still use the complete test dataset. The training data is normalized to 1, and for applying the trained codec we scale the input using a scalar quantized to 8 bits. Another database worth considering for future work is presented in [19]. They recorded multiple free explorations of 81 different materials with a stylus. The materials were grouped in 7 classes, making it especially interesting for classification related research.

An ADAM optimizer was used with an exponential scheduler. The network parameters were initialized randomly and independently sampled for each trained model. The quantization is represented by a Straight-Through Estimator during the backward pass. Its impact on the latent samples is linearly increased by increasing a weighting parameter a_q from 0 to 1 and applying this formula:

$$\hat{\mathbf{y}} = (1 - a_q) \cdot \mathbf{y} + a_q \cdot Q(\mathbf{y}), \tag{9}$$

where $Q(\cdot)$ is the quantization operator. This ensures better gradient flow in the beginning of the training process, while at the end of the training, the quantization takes full effect on $\mathbf{y}$.

We trained different configurations for a bl of 512 to get different Compression Ratios (CRs). Also, different training losses were tested. This led to four deep learning systems under evaluation:

- $L_{P1}, \lambda_1 = 0.9$: trained on MNR-based loss
- $\lambda_1 = 0$: trained on MSE
- $L_{P1}, \lambda_1 = 0.9$, no masking: trained on MNR without masking threshold
- $L_{P2}, \lambda_1 = 0.9$: trained on SMR as weighting coefficients

λ_1 was not set to 1 to not completely remove the influence of the MSE. The loss without masking is signal-independent, since the masking functions are the only component of the globalmask that will change from signal to signal. We trained models with $N = 1$ to 4 downsampling layers as well as varying number of features f and compared their performance. Lower number of features means a smaller size of the latent space and therefore higher compression, so we leveraged this to achieve different compression ratios. Deeper models with a higher downsampling ratio enable us to achieve a smaller latent space and therefore higher compression. In general, the deepest models with four downsampling layers ($N = 4$) performed the best and were used in the evaluation, but did not reach good quality for low CR. Therefore, also two models with only one downsampling layer ($N = 1$) were included, which correspond to the lowest CRs results in the evaluation.

3.2 Metrics Evaluation

The results of our work are compared to the VC-PWQ for different training metric parameters using the SNR as objective metrics and several subjective metrics. The reference to calculate the CR to is 16 bit encoding at the given sampling frequency, which is 2800 Hz. This leads to a raw data rate of 44.8 kbits/s. The SNR is shown in Fig. 5. Overall, we did not get a SNR higher than 35. This could be improved by using more bits in quantization. We set it to 7 bits, which improves the performance for high CR, where we expected the highest impact of the perceptual training loss. The SNR of the version trained with $\lambda_1 = 0$ gets slightly better than the VC-PWQ for most CRs. With L_{P1}, the SNR drops partially for a CR between 5 and 10, which is to be expected, since we optimize a perceptual loss. However, at high CR it is even a little better. The

system without masking performed very similar to the one with masking. The model using L_{P2} did not reach a CR below 5 in our test, but had the highest SNR up to a CR of 20.

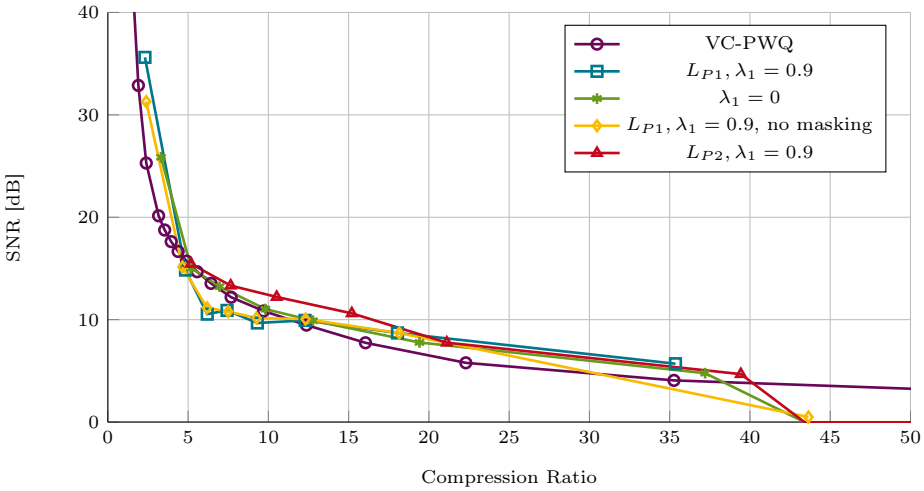


Fig. 5: SNR over Compression Ratio

As in previous work, we used the Spectral Temporal SIMilarity (ST-SIM) as one of the perceptual metrics. It was introduced in [3] and is calculated from spectral and temporal cues. Additionally, we evaluate the MNR using the band structure used for the VC-PWQ. The Spectral Perceptual Quality Index (SPQI) is used for evaluation as well, since it showed better correlation to perceptual tests than the ST-SIM [10]. It computes a perceptually weighted error spectrum and applies a nonlinear function to its sum. We did not use the VibroMAF introduced in [10] since it requires a trained model. A designed metric is better for comparability in general, since it does not depend on the training dataset.

The MNR and SPQI are plotted in Fig. 6 and 7. They show a better performance of the deep learning codec for most of the training losses. The versions trained with L_{P1} showed the highest performance for these metrics with a significant improvement for high CR. The results below CR 20 indicate that having a signal-independent PM for training the models can be sufficient. This could result from the fact that we used the model to optimize the general codec instead of the coding of a specific signal, and it therefore performs better on so far unseen signals with an independent model. The weighted loss achieved better SPQI than the VC-PWQ for most of the range, but overall performed worse than L_{P1}.

The ST-SIM in Fig. 8, on the other hand, had overall lower values for the new codec, even lower ones for the training with L_{P1}. However, L_{P2} reached the best values for the new systems and even higher ST-SIM than the VC-PWQ below

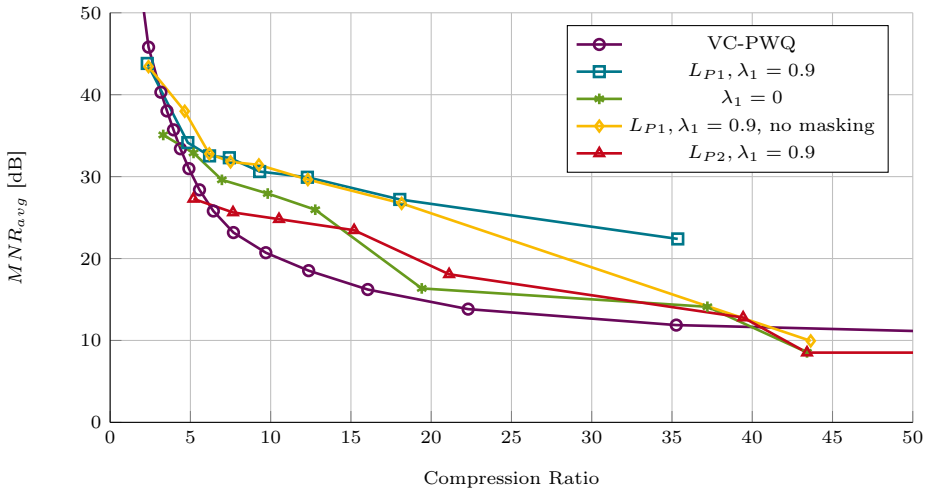


Fig. 6: Averaged MNR over Compression Ratio

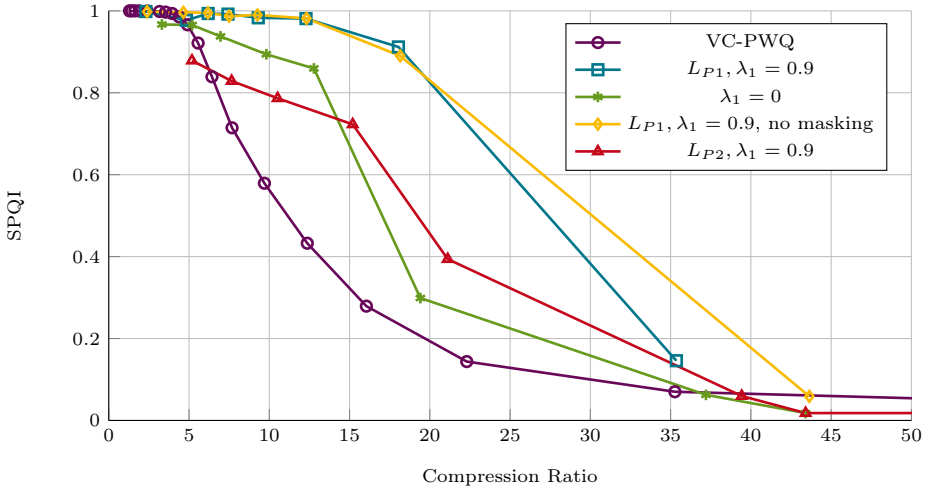


Fig. 7: SPQI over Compression Ratio

a CR of 10. For CR up to 5, the new codec in general has the same ST-SIM as the VC-PWQ or slightly higher. This is acceptable since the SPQI showed clear improvement and was found to correlate better with perceptual experiments [10].

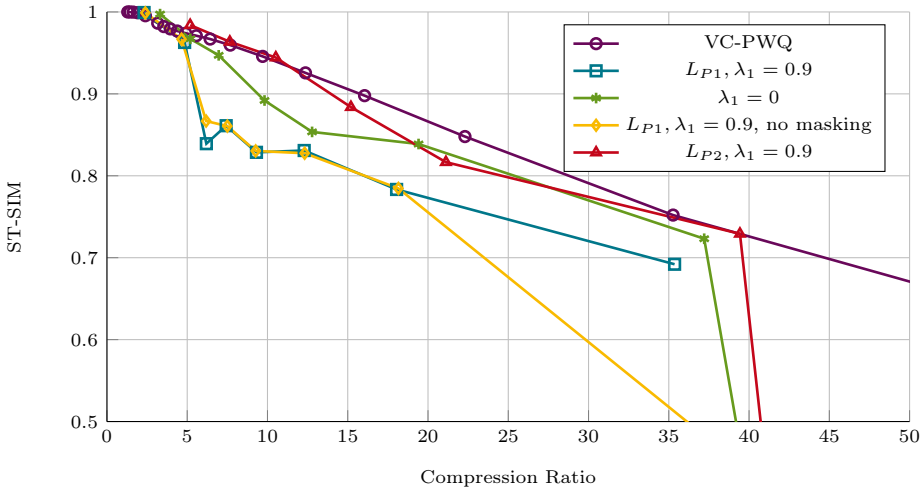


Fig. 8: ST-SIM over Compression Ratio

Since we used random initialization independently for all trained models, improvements introduced by advantageous initialization would not have led to improvements on all models of the same training loss. Also, we trained models with varying hyperparameters N and f for each training loss, which had very consistent improvements in MNR and SPQI for L_{P1}. This indicates that the significant improvements observed in some metrics have their origin in the modified training loss.

Overall, the training using a perceptual loss succeeded in this test, since two of the three perceptual metrics showed a clear improvement while only having a partial decrease in objective quality.

4 Conclusions

In this paper, we introduced an end-to-end learned autoencoder that was successfully trained on a perceptual loss. We adapted the PM that was previously used for the VC-PWQ and surpassed its perceptual performance. We found that using a signal-independent model mostly performed very similar on our test signals to the model that also takes masking effects into account. Using a weighted loss achieved worse MNR than the MNR-based one, so the latter should usually be preferred.

The perceptual metrics used in this work all rely on the human threshold of perception in varying degree, similar to the perceptual losses we developed. This may explain the high performance of the model. However, the SPQI was fit to real perceptual ratings, so a good translation to real applications is to be expected. To fully investigate the improvement on the perceived signal quality, real perceptual tests should be carried out.

The methodology for improved perceptual performance of deep learning-based vibrotactile signal compression is independent from the codec it is applied on and has no impact on the inference process, since only the training loss has to be modified. Therefore, no additional latency is introduced to the real-world application of a codec.

Future work could focus on investigating and improving the quantization of the latent samples to further improve the performance of the codec.

Acknowledgments. Funded by the German Research Foundation (DFG, Deutsche Forschungsgemeinschaft) as part of Germany's Excellence Strategy – EXC 2050/1 – Project ID 390696704 – Cluster of Excellence "Centre for Tactile Internet with Human-in-the-Loop" (CeTI) of Technische Universität Dresden.

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

References

1. Ballé, J., Minnen, D., Singh, S., Hwang, S.J., Johnston, N.: Variational image compression with a scale hyperprior (2018). https://doi.org/10.48550/ARXIV.1802.01436
2. Hamer, R.D., Verrillo, R.T., Zwislocki, J.J.: Vibrotacile masking of pacinian and non-pacinian channels. J Acoust Soc Am. 1983 Apr **73**(4) (1983). https://doi.org/10.1121/1.389278
3. Hassen, R., Steinbach, E.: Subjective Evaluation of the Spectral Temporal SIMilarity (ST-SIM) Measure for Vibrotactile Quality Assessment. IEEE Transactions on Haptics **13**(1), 25–31 (2020). https://doi.org/10.1109/TOH.2019.2962446
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition (2015). https://doi.org/10.48550/arXiv.1512.03385
5. M., Shanmugam, Venusamy, K., S., Subin, S., Srivatsan, O., Naresh Kumar: A comprehensive review of haptic gloves: Advances, challenges, and future directions. In: 2023 Second International Conference on Electronics and Renewable Systems (ICEARS). pp. 227–233 (2023). https://doi.org/10.1109/ICEARS56392.2023.10085607
6. Makous, J., Friedman, R., Vierck, C.: A critical band filter in touch. The Journal of Neuroscience **15**(4), 2808–2818 (Apr 1995). https://doi.org/10.1523/JNEUROSCI.15-04-02808.1995
7. Metzger, A., Toscani, M.: Unsupervised learning of haptic material properties. eLife **11**, e64876 (feb 2022). https://doi.org/10.7554/eLife.64876
8. Nockenberg, L., Noll, A., Panëels, S., Dhiab, A.B., Hudin, C., Steinbach, E.: Mvibcode: Multi-channel vibrotactile codec using hierarchical perceptual clustering. IEEE Transactions on Haptics pp. 1–6 (2023). https://doi.org/10.1109/TOH.2023.3276420

9. Noll, A., Gülecyüz, B., Hofmann, A., Steinbach, E.: A Rate-scalable Perceptual Wavelet-based Vibrotactile Codec. In: 2020 IEEE Haptics Symposium (HAPTICS). pp. 854–859 (2020). https://doi.org/10.1109/HAPTICS45997.2020.ras.HAP20.6.422bbc6e
10. Noll, A., Hofbauer, M., Muschter, E., Li, S.C., Steinbach, E.: Automated Quality Assessment for Compressed Vibrotactile Signals Using Multi-Method Assessment Fusion. In: 2022 IEEE Haptics Symposium (HAPTICS). pp. 1–6 (2022). https://doi.org/10.1109/HAPTICS52432.2022.9765599
11. Noll, A., Nockenberg, L., Gülecyüz, B., Steinbach, E.: Vc-pwq: Vibrotactile signal compression based on perceptual wavelet quantization. In: 2021 IEEE World Haptics Conference (WHC). pp. 427–432 (2021). https://doi.org/10.1109/WHC49131.2021.9517217
12. Pan, D.: A tutorial on MPEG/audio compression. IEEE MultiMedia **2**(2), 60–74 (1995). https://doi.org/10.1109/93.388209
13. Pantera, L., Hudin, C.: Multitouch vibrotactile feedback on a tactile screen by the inverse filter technique: Vibration amplitude and spatial resolution. IEEE Transactions on Haptics **13**(3), 493–503 (2020). https://doi.org/10.1109/TOH.2020.2981307
14. Picinali, L., Feakes, C., Mauro, D., Katz, B.F.: Tone-2 tones discrimination task comparing audio and haptics. In: 2012 IEEE International Workshop on Haptic Audio Visual Environments and Games (HAVE 2012) Proceedings. pp. 19–24. IEEE, Munich, Germany (Oct 2012). https://doi.org/10.1109/HAVE.2012.6374432
15. Richardson, B.A., Vardar, Y., Wallraven, C., Kuchenbecker, K.J.: Learning to feel textures: Predicting perceptual similarities from unconstrained finger-surface interactions. IEEE Transactions on Haptics **15**(4), 705–717 (2022). https://doi.org/10.1109/TOH.2022.3212701
16. Shao, Z., Bao, J., Li, J., Tang, H.: Haptic Recognition of Texture Surfaces Using Semi-Supervised Feature Learning Based on Sparse Representation. Cognitive Computation **15**(5), 1656–1671 (Sep 2023). https://doi.org/10.1007/s12559-023-10141-8
17. Strese, M., Brudermueller, L., Kirsch, J., Steinbach, E.: Haptic Material Analysis and Classification Inspired by Human Exploratory Procedures. IEEE Transactions on Haptics **13**(2), 404–424 (Apr 2020). https://doi.org/10.1109/TOH.2019.2952118
18. Technical University of Munich: Tactile reference data traces, https://cloud.lmt.ei.tum.de/s/pDEPZnGomQYtH4c
19. Toscani, M., Metzger, A.: A database of vibratory signals from free haptic exploration of natural material textures and perceptual judgments (viper): Analysis of spectral statistics. In: Seifi, H., Kappers, A.M.L., Schneider, O., Drewing, K., Pacchierotti, C., Abbasimoshaei, A., Huisman, G., Kern, T.A. (eds.) Haptics: Science, Technology, Applications. pp. 319–327. Springer International Publishing, Cham (2022). https://doi.org/10.1007/978-3-031-06249-0_36
20. Zhao, T., Fang, Y., Wang, K., Liu, Q., Niu, Y.: High Efficiency Vibrotactile Codec Based on Gate Recurrent Network. IEEE Transactions on Multimedia **25**, 5043–5052 (2023). https://doi.org/10.1109/TMM.2022.3186440
21. Zhu, Y.: Perceptual Vibrotactile Signal Code Based on Transformer. In: 2022 IEEE 5th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC). vol. 5, pp. 916–922 (2022). https://doi.org/10.1109/IMCEC55388.2022.10019881

22. Zwicker, E.: Subdivision of the Audible Frequency Range into Critical Bands (Frequenzgruppen). The Journal of the Acoustical Society of America **33**(2), 248–248 (Feb 1961). https://doi.org/10.1121/1.1908630