Noise-to-mask Ratio Loss for Deep Neural Network based Audio Watermarking

Martin Moritz, Toni Olán, Tuomas Virtanen
Tampere University
Tampere, Finland
firstname.lastname@tuni.fi

***Abstract*—Digital audio watermarking consists in inserting a message into audio signals in a transparent way and can be used to allow automatic recognition of audio material and management of the copyrights. We propose a perceptual loss function to be used in deep neural network based audio watermarking systems. The loss is based on the noise-to-mask ratio (NMR), which is a model of the psychoacoustic masking effect characteristic of the human ear. We use the NMR loss between marked and host signals to train the deep neural models and we evaluate the objective quality with PEAQ and the subjective quality with a MUSHRA test. Both objective and subjective tests show that models trained with NMR loss generate more transparent watermarks than models trained with the conventionally used MSE loss**

I. INTRODUCTION

Digital audio watermarking consists in inserting a message into a host audio media, in a transparent way, meaning that the listener should perceive the degradation as little as possible and that their experience should not be affected. The retrieved hidden message can be used to prove the ownership of the media, or for identification and automatic management of the copyrights. The watermarks need to be robust against intentional or unintentional attacks such as compression, filtering or noise addition. The grounds for some traditional techniques like spread spectrum, patchwork, low-bit or phase codings, and echo-based watermarking were set [1] and have been developed further in the following two decades as shown in the review [2] which classifies about 70 solutions for audio watermarking.

In the recent years, some deep neural network (DNN) audio watermarking solutions have emerged. The first [3] is based on two DNN networks: a U-net autoencoder to embed a binary message into the magnitude short time Fourier transform (STFT) of an audio segment and a deep convolutional multi label classifier for the extraction of the predicted message. The phase of the host STFT is directly reused for the watermarked STFT.

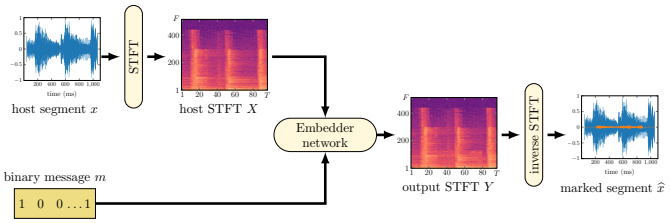
A more recent solution [4] uses similar embedder and extractor networks for speech watermarking, with the important difference that input and output of the embedder are complex-valued STFT and not only their magnitude, and both amplitude and phase of the watermarks are generated.

Those two previous works use the binary cross entropy (BCE) between the input and the extracted messages to guide the learning of the message retrieval, and the mean absolute error (MAE) to minimize the distortion between the input and the marked segments. The embedder and extraction networks are trained jointly and the two losses need to be weighted depending on the epoch: at the beginning the priority is given to the BCE to ensure the extraction of the message and afterwards the weight of the MAE is gradually increased to obtain a better transparency of the watermark. Another recent solution [5] uses an embedder, an extractor and a third discriminator network with an adversarial loss to improve the quality of watermarked audio, but it still uses the mean square error (MSE) loss to minimize the distortion. In case of audio signal, loss functions like MSE or MAE minimize specific statistical criteria which do not match with the human capability to perceive signal modifications introduced by watermarks.

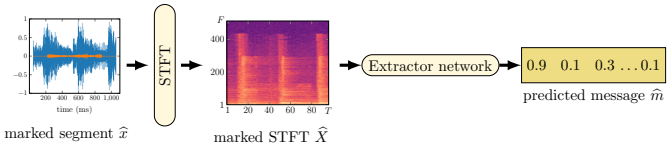
In this paper we propose to use a noise-to-mask ratio (NMR) [6] based loss function for training DNNs, which takes into account the psychoacoustic masking, to improve the perceptual transparency of the audio watermarks. NMR is used as one of the statistics computed from audio signals from which the perceptive evaluation of audio quality (PEAQ) [7] metrics has been calculated, and has been applied for instance to minimize the distortion in non-negative matrix factorization [8].

We do objective and subjective evaluations to show the better transparency of models trained with our novel NMR loss in comparison to the MSE commonly used to optimize the distortion between a DNN generated and an input signals.

We are solely interested in improving the transparency of the watermark thanks to the NMR loss, and we are not addressing the robustness of the watermarking solution. Neither are we dealing with synchronization issue, and for the models we use the watermark needs to be retrieved exactly at the same position that it was embedded. To train a complete audio watermarking system, attacks can be performed on the embedded signals before the extraction. Those attacks can be [4] random noise addition, low-pass filter, or sample suppression. To deal with the synchronization issue, traditional watermarking methods are commonly based on synchronization code [9] which enables to retrieve efficiently the watermark location. In the case of DNN solutions, synchronization attacks can be introduced to train the network to retrieve the message from another location than the one it was exactly embedded. For instance [10] finds efficiently a suitable location by brute force and the usage of pattern bits.



(a) Embedder



(b) Extractor

Fig. 1: The two steps of the watermarking system

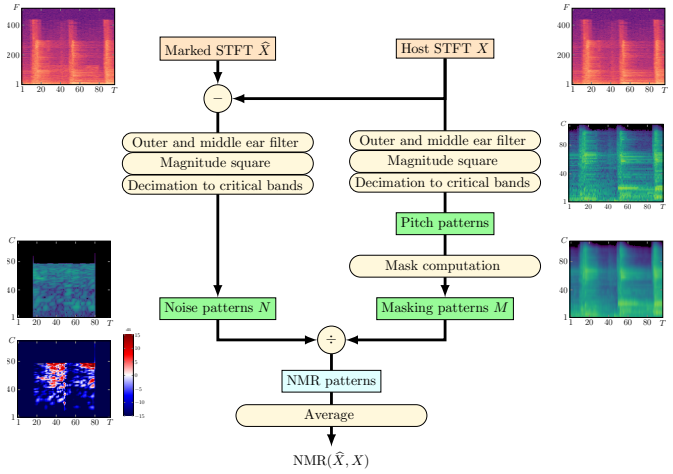


Fig. 2: The computation steps of the NMR

The structure of the paper is as follows: Section II explains the NMR computation and gives the structure of the DNN embedder and extractor for audio watermarking. Section III describes the audio datasets used for the training and the objective and subjective tests, explains how the models are trained and selected, how the tests are performed, and presents the results.

II. Proposed Method

The proposed loss function is applicable with any DNN watermarking architecture but in this study we use an embedder and an extractor network with similar architecture to [4].

The embedder (Fig 1a) takes as input a host audio segment x and a 256-bit binary message m. The STFT X of x is computed and fed with the message to the *embedder network*. The output of the embedder network is the marked STFT Y and the marked time-domain signal $\widehat{x}$ is obtained by its inverse STFT. The extractor (Fig 1b) takes as input the marked segment $\widehat{x}$, computes its STFT $\widehat{X}$ [1] and feeds it to the *extractor network* to get the the sequence of probabilities $\widehat{m}$ which can be rounded to obtain the predicted message of the same length, 256 bits, as the binary message m. The NMR loss is computed between the marked and host STFTs $\widehat{X}$ and X.

All the audio segments have a duration of 1 103 ms and are sampled at $F_S = 44.1\,\text{kHz}$. The discrete Fourier transform (DFT) length is $N = 1024$ samples, and we use the Hanning window and a time-overlap of 50% to compute the STFTs. Therefore the STFT matrices have $F = 513$ rows and $T = 96$ columns, corresponding to the positive DFT frequencies and the STFT time frames, and 2 channels for the real and imaginary parts. The 96-frame duration is chosen to match the decimation operations in the downsampling units of the extractor network.

[1]The STFT Y output by the embedder network and the marked STFT $\widehat{X}$ are not necessarily identical; even though the inverse STFT of Y is $\widehat{x}$, Y might not be the STFT of any time-domain real signal.

A. NMR loss

The proposed loss used for the training of the DNN is a combination of the noise-to-mask ratio (NMR) and the binary cross entropy (BCE) given as

$$\ell(\widehat{X}, \widehat{m}, X, m) = \alpha \text{NMR}(\widehat{X}, X) + (1-\alpha)\text{BCE}(\widehat{m}, m)\,, \quad (1)$$

where α is the transparency weight between 0 (only the extraction matters) and 1 (only the transparency matters).

For computing the NMR in (1), we first need to compute the *noise patterns* $N_{c,t}$, i.e. the squared errors between the marked STFT $\widehat{X}$ and the host STFT X, which are decimated to the $C = 109$ critical band using critical band mapping matrix U, and weighted by outer and middle ear filter ω as

$$N_{c,t} = \sum_{f=1}^{F} U_{c,f} \left| \omega_f (X_{f,t} - \widehat{X}_{f,t}) \right|^2 . \quad (2)$$

Then the NMR is the average of the ratios of the noise patterns and the *masking patterns* $M_{c,t}$ of the host signal:

$$\text{NMR}(\widehat{X}, X) = \frac{1}{C \cdot T} \sum_{c=1}^{C} \sum_{t=1}^{T} \frac{N_{c,t}}{M_{c,t}} . \quad (3)$$

Above, c is the critical band index, f is the STFT frequency index, and t is the time frame index.

The detailed processing steps [7] of NMR calculation are illustrated on Fig. 2 and summarized are as follows. 1) Both complex-valued STFT X and $\widehat{X}$ are computed and scaled by a loudness calibration constant as some factors of the model are level dependent. 2) The error $\widehat{X} - X$ is weighted for each DFT frequency by the frequency response ω_f of the outer and middle ear filter and the magnitudes are squared. 3) These squared errors are decimated according to the 109 critical bands to get the noise patterns $N_{c,t}$. The decimation is obtained by a multiplication by the DFT-to-critical bands mapping matrix U. The critical frequency bands range from 80 Hz to 18 000 Hz with a constant resolution of 0.25 Bark. The rows

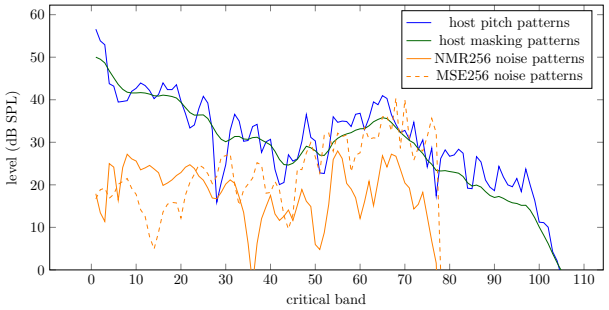


Fig. 3: Pitch and masking patterns for a host signal, and the noise patterns for marked signals generated by two (NMR256 and MSE256) of our models

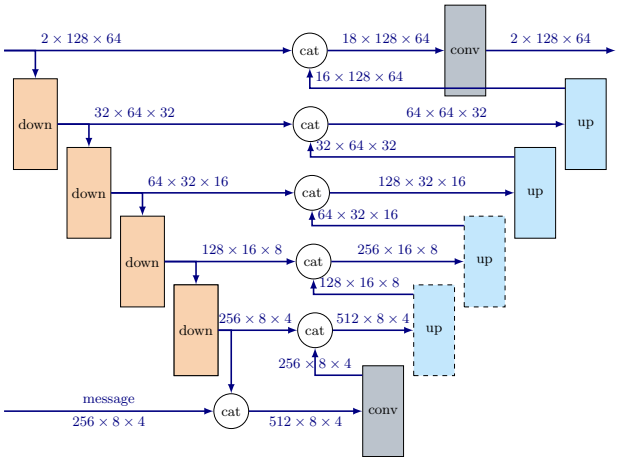


Fig. 4: The embedder network

of the matrix U are the magnitude responses of band-pass filters equal to one inside the critical band, zero outside of it, and linearly interpolated in the transition bands. The columns of the matrix U sum up to one. 4) The host STFT is itself similarly scaled by the loudness calibration constant, weighted by ω and the squared magnitude is decimated to the critical bands with the multiplication by the DFT-to-critical bands mapping matrix U to result into the *pitch patterns*. Those pitch patterns are spread first over the frequencies to model simultaneous masking and then in time to model temporal masking to get the excitation patterns, which are multiplied by a frequency-dependent offset to get the *masking patterns* $M_{c,t}$. The masking patterns correspond to the masking thresholds below which a sound would be imperceptible. 5) The NMR is the average of the ratios (3) of the noise patterns $N_{c,t}$ and the masking patterns $M_{c,t}$. The pitch and the masking patterns of a host signal in one example time frame, and the noise patterns between the host and marked signals for two models that we trained are shown on Fig. 3.

The NMR calculation process is detailed in [7] and some clarifications are given by [11] where (3), converted into dB, is referred as the total NMR. In the ITU-R recommendations [7], the error to compute the noise patterns is the squared difference of the magnitudes but here we compute the squared magnitude of the complex difference as shown in (2). The reason for that lies in the fact that we want to use the NMR as a loss to generate a marked signal and we need not only the amplitude but also the phase to be as close as possible to the host signal. The sampling rate and STFT parameters we use are different than in [7] where the signal is sampled at 48 kHz, the length of the DFT is 2048 samples.

B. The Embedder

A 128×64-submatrix of the host STFT X, cropped between the 4^{th} and 131^{st} frequencies and between the 17^{th} and the 80^{th} time frame, is fed to the embedder network. The goal of the frequency cropping is to limit the watermark in a frequency region (80 Hz-5.6 kHz) containing most of the audio content so that the watermark cannot be removed by simple filtering. The cropping in time was intended to improve the extraction of a desynchronized watermark (which is not addressed in this study). As a consequence of it, there is no watermark at the beginning and at the end of the marked segments as shown on Fig. 1.

As in [4] the embedder network (Fig. 4) is based on the U-net model and is composed of an encoder and a decoder. The encoder contains four downsampling units, each consisting of a convolutional layer, a 2D batch normalization layer and a leaky ReLU activation. Halving the width and the height of the feature maps is obtained with a stride of 2×2 in the convolutional layers. The amount of channels is increased from 2 in the input to 256 in the bottommost layer where the feature map has a size of 8×4. This last output is concatenated along the channels dimension with the binary message where each of the 256 bit is replicated 8×4 times to constitute 256 channels. This 512-channel concatenation is fed to an embedding unit composed of a convolutional layer of stride 1×1 followed by a leaky ReLU activation, which outputs a feature map of 256 channels.

The decoder contains four upsampling units with skip connections coming from the encoder at each level. The input of the upsampling units are the concatenation of the output of the corresponding downsampling unit and the output of the unit below. The upsampling units are composed of a transposed convolution with a stride 2×2, a 2D batch normalization layer and a ReLU activation. For the two bottommost ones, the activation is followed by a dropout layer of probability 50% as proposed in [4] to increase the robustness of the system against attacks. The output of each upsampling unit has the same size (number of channels, height and width) than the input of the downsampling unit at the same level at the exception of the topmost unit which has 16 channels, whereas the network input feature map has two. Finally, those two feature maps are concatenated and fed to a convolutional layer. This upper skip connection is added in comparison to [4], and another difference is that we have four downsampling and upsampling units whereas the embedder in [4] has one more of each and an intermediary feature map of 16 channels.

The full output STFT Y is obtained by inserting

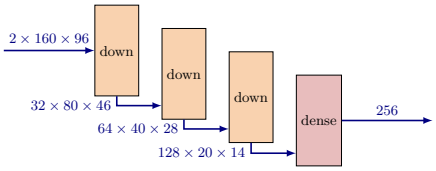


Fig. 5: The extractor network

the $2 \times 128 \times 64$ autoencoder output into the host STFT between the 4^{th} and 131^{st} frequency, and the 17^{th} and 80^{th} time frame. The inverse STFT of Y is returned by the embedder as the marked segment $\widehat{x}$.

C. The Extractor

The 160 lowest frequency bins and 96 time frames from the STFT $\widehat{X}$ of the marked segment are fed to the *extractor network* (Fig. 5). There is no cropping in time to give the possibility to recover a desynchronized watermark.

The *extractor network* contains three downsampling units, each composed of a convolutional layer with a stride of 2×2, a 2D batch normalization layer, and a leaky ReLU activation. The final unit's output is fed to a dense layer and a sigmoid activation to compute the predicted message: a sequence of 256 floats, between 0 and 1.

In both embedder and extractor, the convolutional and transposed convolutional layers have kernels of size 5×5, and the ones followed by a batch normalization have no bias. The leaky ReLUs have a negative slope of 0.2.

We also designed a second model to embed messages of length 512 bits. The only differences are the number of input channels of the convolutional layer in the embedding unit (768 instead of 512) and the dimension for the output of the dense layer in the extractor (512 instead of 256).

III. Evaluation

We evaluate two models with message lengths 256 and 512 bits. Each is trained in two versions, one with the loss (1) that uses the NMR, and the second one where MSE is used instead of the NMR loss.

A. The audio data

The models are trained with a subset of the FMA audio dataset (medium version) [12]. The original dataset contains 25 000 tracks of 30 s, in mp3 format from 16 unbalanced genres, at a sampling rate of 44.1 kHz. The tracks which original bit rate compression is lower than 192 kbit/s are discarded. Each randomly selected track is converted to mono, cropped, and split into twenty-four 1 103 ms-long segments for which the masking patterns are computed. Those 24 segments are put into the same randomly selected training or validation set to reach a total duration of respectively 80 and 10 hours.

B. Training

A set of models is trained with MSE instead of NMR in the loss formula (1). For a message length of 256 bits, an initial model is trained for one epoch with weight $\alpha = 0.5$. Then the weight is given its final value and the training resumed for 29 more epochs. We choose five different final values for α (0.8, 0.9, 0.92, 0.95, and 0.98) and train three models for each of them; if we use directly the final weight values, without the initial epoch, the model would extract random messages. Those final values for α were chosen to get a set of models with various and decent acoustic properties and accuracy for the extraction so that all of them could be selected for the listening tests.

A second initial model is trained with the NMR for the three first epochs with successive weights $\alpha = 10^{-4}$, 10^{-2} and 10^{-1}. Then three models are trained with each of the final values for α (0.1, 0.2, 0.3, 0.4, and 0.5) for 27 more epochs. Those final weights values are chosen to get 15 NMR trained models with bit error rate (BER) comparable to the 15 previously MSE trained models.

Except the initial and final weights, and the distortion loss, all the models, with identical architecture, are trained in the same conditions (learning rate, batch size, Adam optimizer). The binary message is randomly drawn for each audio segment and at each epoch. The training sessions lasted at most 30 epochs, with early stopping when the validation loss did not improve for two consecutive epochs to prevent overfitting. The model at the epoch with the best validation loss ℓ is selected. The balance between the accuracy of the message extraction and the transparency of the watermark depends on the final value of the weight α. For two models trained with the same loss function, a higher final value of α generally means a better transparency (measured with MSE or NMR) and a lower accuracy (measured with the BCE).

Out of those 15 models trained with MSE, we select the one with the best signal-to-noise ratio (SNR) on the validation set. Out of the 15 NMR trained models, we select the one with the best NMR among those with lower BER than the selected MSE model. Those two models are the MSE256 and the NMR256 models of the Table I.

The same scenario is reproduced with an input message of 512 bits, and we obtain the four models of the Table I. We can observe that for this longer messages the BER is becoming very high and the SNR slightly worse and those models would not be accurate enough to be used for efficient audio watermarking.

TABLE I: The four selected trained models. The validation metrics used for the selection are in blue

Model	BER	SNR (dB)	NMR (dB)
MSE256	0.00030	25.8	13.7
NMR256	0.00025	22.4	−18.0
MSE512	0.11100	25.0	13.7
NMR512	0.09700	18.6	−13.3

C. Objective evaluation

The objective evaluation is performed with the 100 tracks of the Music Genre Database [13]. For each track thirty 1103 ms-long segments are randomly selected. The PEAQ [7] is evaluated between the segments marked with each four

selected models and the host segments. We used the MATLAB implementation by Kabal [11]. The results are presented on Fig. 6.

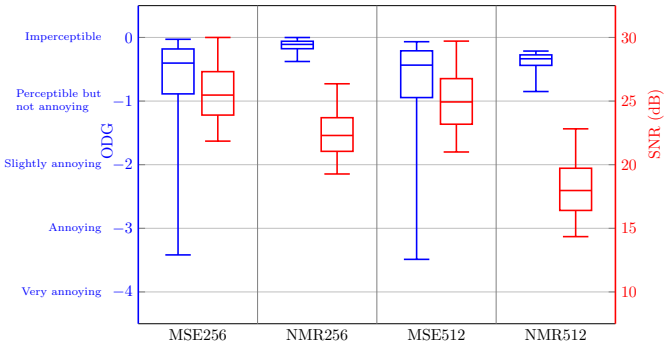


Fig. 6: PEAQ (objective difference grade (ODG) scale) in blue. The red boxes represent the SNR. The upper and lower whisker corresponds to the 5$^{\text{th}}$ and the 95$^{\text{th}}$ percentiles, the outliers are not shown

With the NMR-models more than 95% of the 3000 segments get a PEAQ rating between 0 (*imperceptible*) and -1 (*perceptible but not annoying*). For the MSE-models around 25% of the segments are rated between -1 and -4 (*very annoying*). The segments marked by the two NMR-models have a lower SNR in comparison to the ones marked by the MSE models, even though they have a better objective perceptual quality.

D. Subjective evaluation

We organized a MUSHRA test [14] with the platform WEBMushra [15] to evaluate the audio quality of the models of the Table I. Although those test have been initially designed to assess audio coding they have been used to evaluate the quality of watermarked audio [16], [17] as well.

We selected manually the 7 clips of 10 s shown on Table II from the tracks used in the objective tests with the idea to present a variation of musical genre, number of instruments and spectral signature. For each clip the assessors had to evaluate blindly 7 versions: the original clip, two low-pass filtered anchors at 3.5 and 7 kHz (recommended in MUSHRA tests), and the versions marked with the 4 trained models where the original clip is padded with zeros to the duration of 10 segments, the segments are embedded, concatenated and the result is cropped to the original 10 s-duration. The 20 assessors were naive listeners, between 15 and 50 years old, and had no previous experience in evaluation of audio quality. The results are shown on Fig. 7.

TABLE II: The 10-second clips used for the MUSHRA test

Study ID	Filename	Title	Start time (s)
pop	G001	Wasting Time	240
techno	G021	Stf	20
jazz	G030	Kitchen	58
orchestra	G050	Egmont	210
chanson	G087	Je te Veux	93
flamenco	G082	Sevillanas	70
piano	G059	Rondo in D Major	20

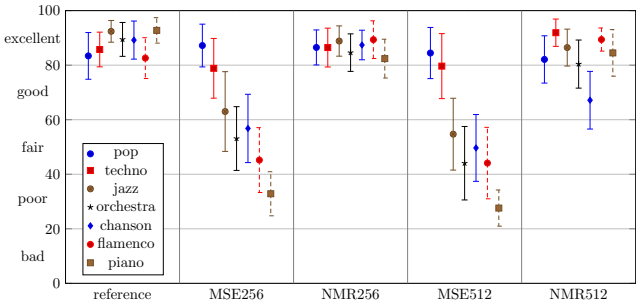


Fig. 7: Mean and 95% confidence intervals of the MUSHRA scores for the 7 clips and their versions marked with the four trained models

For thirteen of the fourteen clips marked with the NMR-models the average rate is *excellent* (between 80% and 100%) whereas it is the case for only four of the clips marked with the MSE-models. For those, the two original clips (*pop* and *techno*) are acoustically busy with multiple instruments playing at the same time. For the *chanson* marked with the NMR512-model the quality was rated as *good* showing that the watermark is not transparent, but the subjective evaluation is better than the *fair* grade given for both MSE-models. The worse ratings have been given to mono-instrumental clips (guitar and piano) marked with MSE-models.

IV. Conclusion

In this study we propose a perceptual loss based on the NMR which models the psychoacoustic masking effect of the human ear. We use this loss to train an autoencoder DNN for music watermarking in the goal of minimizing the perceptual distortion between the host and the marked signal. We also train from scratch the same encoder with NMR replaced by the commonly used MSE loss and we compare objective quality with PEAQ and subjective quality with MUSHRA tests. Both tests show clear superiority of the audio quality of the watermarked audio embedded by models trained with our perceptual NMR loss. Our solution is not dealing with robustness and synchronization, and this could be addressed by further studies where a DNN watermarking system could be trained to resist to attacks and some mechanism found to tackle the synchronization issue. Our NMR loss would be used in those future solutions to improve the transparency of the generated watermarks.

References

[1] W. Bender, D. Gruhl, N. Morimoto, , and A. Lu, "Techniques for data hiding," in *IBM Systems Journal*, vol. 35, no. 3-4, 1996.

[2] G.Hua, J.Huand, Y.Q.Shi, J.Goh, and V.L.Thing, "Twenty years of digital audio watermarking – a comprehensive review," in *Signal Processing*, vol. 128, 2016, pp. 222–242.

[3] L. Tegendal, "Watermarking in audio using deep learning," Master's thesis, Dept. Comput. Vision, Linköping University, Linköping, Sweden, 2019.

[4] K. Pavlović, S. Kovačević, I. Djurović, and A. Wojciechowski, "Robust speech watermarking by a jointly trained embedder and detector using a dnn," in *Digital Signal Processing*, vol. 122, 2022, p. 103381.

[5] C. Liu, J. Zhang, H. Fang, Z. Ma, W. Zhang, and N. Yu, "DeAR: A deep-learning-based audio re-recording resilient watermarking," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 11, 2023, pp. 13 201–13 209.
[6] K. Brandenburg, "Evaluation of quality for audio encoding at low bit rates," in *Audio Engineering Society Convention 82*, 1987.
[7] *Method for objective measurements of perceived audio quality*, ITU-R recommendation BS.1387, 1998.
[8] J. Nikunen and T. Virtanen, "Noise-to-mask ratio minimization by weighted non-negative matrix factorization," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pp. 25–28.
[9] S. Wu, J. Huang, D. Huang, and Y. Shi, "Efficiently self-synchronized audio watermarking for assured audio data transmission," *IEEE Transactions on Broadcasting*, vol. 51, no. 1, pp. 69–76, 2005.
[10] G. Chen, Y. Wu, S. Liu, T. Liu, X. Du, and F. Wei, "Wavmark: Watermarking for audio generation," *arXiv.org*, 2023.
[11] P. Kabal, "An examination and interpretation of ITU-R BS.1387:perceptual evaluation of audio quality," Dept. of Elect. Comput. Eng., McGill Univ., Montréal, Quebec, Rep., 2002.
[12] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, "FMA: A dataset for music analysis," in *18th International Society for Music Information Retrieval Conference*, 2017.
[13] M. Goto, "Development of the RWC music database," in *18th International Congress on Acoustics*, vol. 1, 2004, pp. 553–556.
[14] *Method for the subjective assessment of intermediate quality level of audio systems*, ITU-R recommendation BS.1534, Rev. 3, 2015.
[15] M. Schoefler, "webMUSHRA — a comprehensive framework for web-based listening tests," in *Journal of Open Research Software*, vol. 6, 2018.
[16] R. Tachibana, "Sonic watermarking," in *EURASIP Journal on Advances in Signal Processing*, no. 13, 2004.
[17] K. Kondo, "On the use of objective quality measures to estimate watermarked audio quality," in *2012 Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2012, pp. 126–129.