electronics

MDPI

Article

Stochastic Restoration of Heavily Compressed Musical Audio Using Generative Adversarial Networks

Stefan Lattner * and Javier Nistal

Sony Computer Science Laboratories (CSL), 75005 Paris, France; javier.nistalhurle@sony.com
* Correspondence: stefan.lattner@sony.com or me@stefanlattner.at

Abstract: Lossy audio codecs compress (and decompress) digital audio streams by removing information that tends to be inaudible in human perception. Under high compression rates, such codecs may introduce a variety of impairments in the audio signal. Many works have tackled the problem of audio enhancement and compression artifact removal using deep-learning techniques. However, only a few works tackle the restoration of *heavily compressed* audio signals in the *musical domain*. In such a scenario, there is no unique solution for the restoration of the original signal. Therefore, in this study, we test a *stochastic* generator of a Generative Adversarial Network (GAN) architecture for this task. Such a stochastic generator, conditioned on highly compressed musical audio signals, could one day generate outputs indistinguishable from high-quality releases. Therefore, the present study may yield insights into more efficient musical data storage and transmission. We train stochastic and deterministic generators on MP3-compressed audio signals with 16, 32, and 64 kbit/s. We perform an extensive evaluation of the different experiments utilizing objective metrics and listening tests. We find that the models can improve the quality of the audio signals over the MP3 versions for 16 and 32 kbit/s and that the stochastic generators are capable of generating outputs that are closer to the original signals than those of the deterministic generators.

Keywords: musical audio enhancement; audio restoration; bandwidth extension; compression artifact removal; codec enhancement; generative adversarial network

Citation: Lattner, S.; Nistal, J. Stochastic Restoration of Heavily Compressed Musical Audio Using Generative Adversarial Networks. *Electronics* **2021**, *10*, 1349. https://doi.org/10.3390/electronics10111349

Academic Editors: Alexander Lerch and Peter Knees

Received: 30 March 2021
Accepted: 1 June 2021
Published: 5 June 2021

1. Introduction

The introduction of MP3 (i.e., MPEG-1 layer 3 [1]) was transformative in how music was stored, transmitted, and shared in digital devices and on the internet. MP3 players, sharing platforms and streaming resulted directly from the possibility to considerably compress audio data without noticeable perceptual compromises. Compared to *lossless* audio coding formats, which allow for a perfect reconstruction of the original PCM audio signal, *lossy* formats (like MP3) typically lead to better compression by ignoring the parts of the signal to which humans are less sensitive. This process is also called *perceptual coding*, which takes into account the physio- and psychological abilities of human auditory perception, resulting in so-called psychoacoustic models [2].

While there exist several different lossy audio codecs (e.g., AAC, Opus, Vorbis, AMR), MP3 is undoubtedly the most commonly used. It is built upon an analysis filter bank and a subsequent computation of the modified discrete cosine transform (MDCT). In parallel, the signal is analyzed based on a perceptual model that exploits the psychoacoustic phenomena of *auditory masking* to determine sound events in the audio signal that are considered to be beyond human hearing capabilities. Based on this information, the spectral components are quantized with a specific resolution and coded with variable bit allocation, while keeping the noise introduced in this process below the masking thresholds [3]. This process may introduce a variety of deficiencies when configured with incorrect or very extreme parameters. For example, under large compression rates, high-frequency content is susceptible to being removed, resulting in *bandwidth loss*. *Pre-echoes* can occur when decoding very sudden sound events for which the quantization noise spreads

out over the synthesis window and consequently precedes the event causing the noise. Other common artifacts are so-called *swirlies* [4], characterized by fast energy fluctuations in the low-level frequency content of the sound. Furthermore, there are other problems related to MP3 compression, such as *double-speak*, as well as a general loss of transient definition, transparency, loss of detail clarity, and more [4].

Many works exist which tackle the problem of audio enhancement, including the removal of compression artifacts. The most common recent methods used for these types of problems are based on deep learning. Typically, they focus on specific types of impairments present in the audio signals (e.g., reverberation [5], bandwidth loss [6], or audio codec artifacts [7–11]). Additionally, different types of neural network architectures have been studied for these tasks; for example, Convolutional Neural Networks (CNNs) [12], WaveNet-like architectures [8,13], and UNets [14,15]. However, most of the works in this line of research tackle the enhancement of *speech* signals [7–10,12–18], and only a few publications exist for *musical* audio restoration [11,19–21]. This focus on speech is understandable, given the wide range of speech enhancement techniques in telephony, automatic speech recognition, and hearing aids. Additionally, compared to musical audio signals, speech signals are easier to study, as they are more homogeneous, narrow-banded, and usually monophonic. In contrast, musical audio signals, particularly in the popular music genre, are highly varied. They typically consist of multiple superimposed sources, which can be of any type, including (polyphonic) tonal instruments, percussion, (singing) voice, and various sound effects. In addition, music is typically broad-banded, containing frequencies spanning over the entire human hearing range.

Given that studies on deep-learning-driven audio codec artifact removal for musical audio data are underrepresented in audio enhancement research, in this work, we attempt to provide some more insights into this task. We investigate the limits of a generative neural network model when dealing with a general popular music corpus comprising music released in the last seven decades. In particular, we are interested in the ability of the model to regenerate lost information of heavily compressed musical audio signals using a *stochastic generator* (which is not very common in audio enhancement, with [10,22] being some exceptions). This work is not only relevant for the restoration of MP3 data in existing (older) music collections. In light of current developments in musical audio generation, where full songs can already be generated from scratch [23], musical audio enhancement may soon possess a much more *generative* aspect. It has already been shown that strong generative models can enhance heavily corrupted speech through resynthesis with neural vocoders [22]. Along these lines, examining a *generative* (i.e., stochastic) decoder for heavily compressed audio signals may contribute to insights about more efficient musical data storage and transmission. Today, music streaming is increasingly common, which poses issues regarding energy consumption, and consequently, environmental sustainability. When accepting deviations from the original recording, higher compression rates could be reached with a generative decoder without perceptual compromises in the listening experience. Moreover, for heavily compressed audio signals, there is no single best solution to recover the original version. Therefore, it may be interesting for users to generate multiple recoveries and pick the one they like most.

We introduce a Generative Adversarial Network (GAN) [24] architecture for the restoration of MP3-encoded musical audio signals. We train different stochastic and deterministic generators on MP3s with different compression rates. Using these models, we investigate whether (1) restorations of the models considerably improve the MP3 versions, (2) if we can systematically pick samples among the outputs of the stochastic generators which are closer to the original than such of the deterministic generators, and (3) if the stochastic generators generally output higher-quality restorations than the deterministic generators. To that end, we perform an extensive evaluation of the different experiment setups utilizing objective metrics and listening tests. We find that the models are successful in points 1 and 2, but the *random* outputs of the stochastic generators are approximately on a par with (i.e., do not improve) the overall quality compared to the deterministic models (point 3).

The proposed GAN architecture is based on dilated convolutions with skip connections, combined with a novel concept which we call Frequency Aggregation Filters. These are convolutional filters spanning the whole frequency range, which contribute to the stability of the training and constitute a consequent take on the problem of non-local correlations in the frequency spectrum (see Section 3.1.3). We also find that using so-called self-gating considerably reduces the memory requirement of the architecture by halving the number of input maps to each layer without degradation of the results (see Section 3.1.2). In order to prevent mode collapse, we propose a regularization that enforces a correlation between differences in the noise input and differences in the model output (see Section 3.2.1). As opposed to most other works (but in line with a few other approaches using GANs [25] and U-Net-based architectures [14,15]), we directly input (and output) the (non-linearly scaled) complex-valued spectrum to the generator, eliminating the need to deal with phase information separately.

The rest of this paper is organised as follows. In Section 2 we revise previous works in bandwidth extension and audio enhancement. In Section 3 we describe in depth the proposed GAN architecture (Section 3.1), the training procedure (Section 3.2), the dataset (Section 3.3) and the evaluation methods (Section 3.4). Finally, in Section 4 we present and discuss the results and conclude with suggestions for future work in Section 5. Audio examples of the work are provided in the accompanying website (Available online: https://sonycslparis.github.io/restoration_mdpi_suppl_mat/ (accessed on 4 June 2021).

2. Related Work

In this work, Generative Adversarial Networks (GANs) are employed to restore MP3-compressed musical audio signals to their original high-quality versions. This task falls into the intersection of audio enhancement and bandwidth extension. Therefore, we review works on both these domains.

2.1. Bandwidth Extension

Low-resolution audio data (i.e., audio signals with a sample rate lower than 44.1 kHz) are generally preferable for storage or transmission over band-limited channels, like streaming music over the internet. Additionally, lossy audio encoders can significantly reduce the amount of information by removing high-frequency content, but at the expense of potentially hampering the perceived audio quality. In order to restore the quality of such truncated audio signals, bandwidth extension (BWE) methods aim to reconstruct the missing high-frequency content of an audio signal given its low-frequency content as input [26]. BWE is alternatively referred to as *audio re-sampling* or *sample-rate conversion* in the field of Digital Signal Processing (DSP), or as *audio super-resolution* in the Machine Learning (ML) literature. Methods for BWE have been extensively studied in areas like audio streaming and restoration, mainly for legacy speech telephony communication systems [13,16,17,27] or, less commonly, for degraded musical material [19,20].

Pioneering works to speech BWE were originally algorithmic and operated based on a source-filter model. There, the problem of regenerating a wide-band signal is divided into finding an upper-band source and the corresponding spectral envelope, or filter, for that upper band. While methods for source generation were based on simple modulation techniques, such as spectral folding and translation of a so-called low-resolution baseband [28], the efforts focused on estimating the filter or spectral envelope [29]. These works introduced the so-called spectral band replication (SBR) method, where the lower frequencies of the magnitude spectra are duplicated, transposed, and adjusted to fit the high-frequency content. Because in most use-cases for speech BWE the full transmission stack is controlled, most of these algorithmic methods rely on side information about the spectral envelope, obtained at the encoder from the full wide-band signal, and then transmitted within the bitstream for subsequent reconstruction at the decoder.

Learning-based approaches to speech BWE rely on large models to learn dependencies across the lower and higher ends of the frequency spectrum. Methods based on non-

negative matrix factorization (NMF) treat the spectrogram as a fixed set of non-negative bases learned from wide-band signals [27]. These bases are fixed at test time and used to estimate the activation coefficients that best explain the narrow-band signal. The wide-band signal is then reconstructed by a linear combination of the base vectors weighted by the activations. These methods efficiently up-sample speech audio signals up to 22.05 kHz, but are sensitive to non-linear distortions due to the linear-mixing assumption. Dictionary-based methods can significantly improve the speech quality over the NMF approach by reconstructing the high-resolution audio signals as a non-linear combination of units from a pre-defined clean dictionary [30], or by casting the problem as an l1-optimization of an analysis dictionary learned from wide-band data [31].

Early works on speech BWE using neural networks inherited the source-filter methodology found in previous works. By employing spectral folding to regenerate the wide-band signal, a simple NN is used to adjust the spectral envelope of the generated upper-band [16]. Direct estimation of the missing high-frequency spectrum was not extensively studied until the introduction of deeper architectures [17]. Advances in computer vision [32,33] inspired the usage of highly expressive models to audio BWE, leading to significant improvements in the up-sampling ratio and quality of the reconstructed audio signal. Different approaches followed: by generating the missing time-domain samples in a process analogous to image super-resolution [34], by inpainting the missing content in a time-frequency representation [20], or by combining information from both domains, preserving the phase information [35]. Powerful auto-regressive methods for raw audio signals based on SampleRNN [36] or WaveNet [13] are able to increase the maximum resolution to a 16 kHz and 24 kHz sample-rate, respectively, without neglecting phase information, as it is the case in most works operating in the frequency domain [6,17,19,20,27]. Most recent techniques using sophisticated transformer-based GANs can up-sample speech to full-resolution audio at a 44.1 kHz sample rate [6].

2.2. Audio Enhancement

Audio signals may suffer from a wide variety of environmental adversities, for example: sound recordings using low-fidelity devices or in noisy and reverberant spaces; degraded speech in mobile or legacy telephone communications systems; musical material from old recordings, or heavily compressed audio signals for streaming services. Audio enhancement aims to improve the quality of corrupted audio signals by removing noisy additive components and restoring distorted or missing content to recover the original audio signal. The field was first introduced for applications in noisy communication systems to improve the quality and intelligibility of speech signals [37]. Many studies have been carried out on speech audio enhancement, such as for speech recognition, speaker identification and verification [38–40], hearing assistance devices [41,42], de-reverberation [5], and so on. In the specific case of audio codec restoration, many different techniques exist for improvement of speech signals [7–10], yet only few works have attempted the restoration of heavily compressed *musical* audio signals [11,21].

Classic speech enhancement methods follow multiple approaches, primarily based on analysis, modification, and synthesis of the noisy signal's magnitude spectrum and often omitting phase information. Popular strategies are categorized into spectral subtraction methods [43], Wiener-type filtering [44], and statistical model-based [45] and subspace methods [46]. These approaches have proven successful when the additive noise is stationary. However, under highly non-stationary noise or reduced signal-to-noise ratios (SNR), they introduce artificial residual noise.

Recent deep-learning approaches to speech enhancement outperform previous methods in terms of perceived audio quality, effectively reducing both stationary and non-stationary noise components. Popular methods learn non-linear mapping functions of noisy-to-clean spectrogram signals [18] or learn masks in a time-frequency domain representation [5,14,47]. Many architectures have been proposed: basic feed-forward DNNs [18], CNN-based [12], RNN-based [48], and more sophisticated architectures based

on WaveNet [8] or U-Net [14]. GANs are also increasingly popular in speech enhancement [49–52]. Pioneering works using GANs operated either on the waveform domain [49] or on the magnitude STFT [53]. Subsequent works mainly focused on the latter representation due to the reduced complexity compared to time-domain audio signals [51,52,54]. Recent works operating directly on the raw waveform were able to consider a broader type of signal distortions [50] and to improve the reduction of artifacts over previous works [55]. Successive efforts were made to further reduce artefacts by, for example, taking into consideration human perception. Some works directly optimize over differentiable approximations of objective metrics, such as PESQ [54]. However, these metrics correlate poorly with human perception, and some works defined the objective metric in embedding spaces from related tasks [56] or by matching deep features of real and fake batches in the critic's embedding space [57].

The vast majority of the speech audio enhancement approaches mentioned above operate on the magnitude spectrum and ignore the phase information [20,21,51,52]. At synthesis, researchers often reuse the phase spectrum from the noisy signal, introducing audible artifacts that would be particularly annoying in musical audio signals. To address this, phase-aware models for speech enhancement use a complex ratio mask [47], or, as we have seen, operate directly in the waveform domain [50,55]. Inspired by a recent work demonstrating that DNNs implementing complex operators [58] may outperform previous architectures in many audio-related tasks, new state-of-the-art performances were achieved on speech enhancement using complex representations of audio data [14,15]. Recent work was able to further improve these approaches by introducing a complex convolutional block attention module (CCBAM) and a mixed loss function [59].

3. Materials and Methods

In the following, we describe the experiment setup. This includes the model architecture (see Section 3.1) and the training procedure (see Section 3.2). Furthermore, the data used and the data representation are presented in Section 3.3, and the objective and subjective evaluation methods are discussed in Section 3.4.

3.1. Model Architecture

The model employed in this work is a Generative Adversarial Network (GAN), conditioned on spectrogram representations of MP3-compressed audio files (see Figure 1 for an overview on architecture and training). As common in GANs, there are two separate models, the generator G and the critic D. G receives as input an excerpt of an MP3-compressed musical audio signal in spectrogram representation y (i.e., non-linearly scaled complex STFT components, see Section 3.3.1) and learns to output a restored version $\hat{x}$ of that excerpt (i.e., the fake data), approximating the original, high-quality signal x. D learns to distinguish between such restorations $\hat{x}$ and original high-quality versions of the signal x (i.e., the true data). In addition to the true/fake data, D also receives the MP3 versions of the respective excerpts. That way, it is ensured that the information present in the MP3 data is faithfully preserved in the output of G. We test stochastic and deterministic generators in our experiments. For the stochastic models, we also provide some noise input $z \sim \mathcal{N}(0, \mathbf{I})$, resulting in different restorations for a given MP3 input.

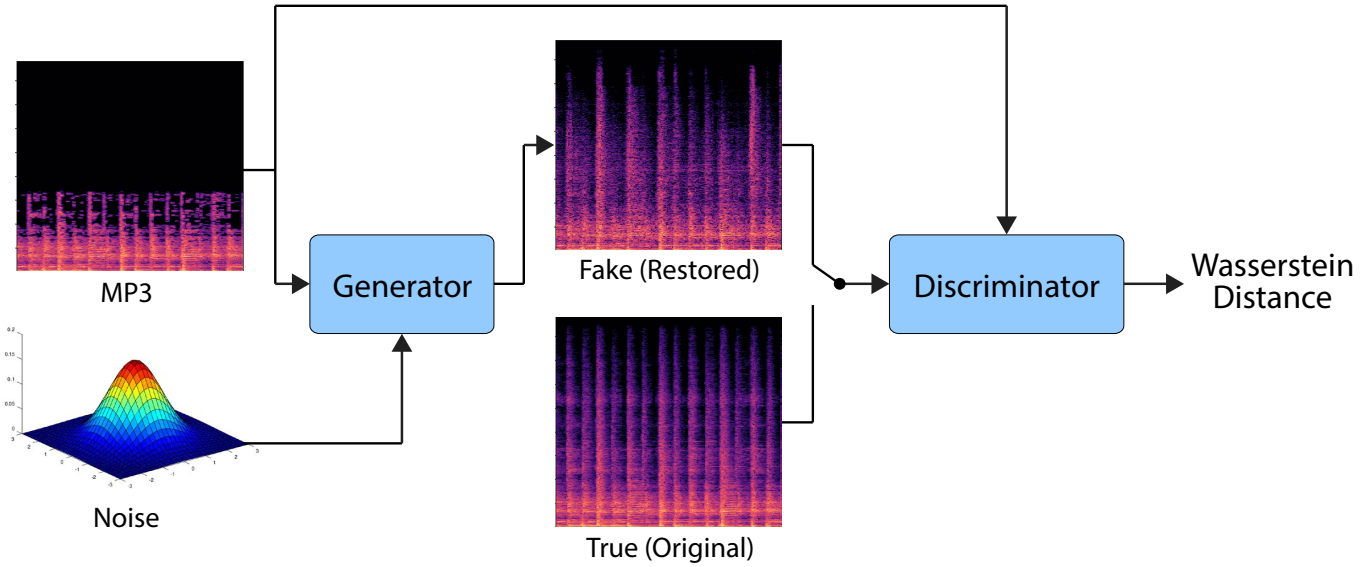


Figure 1. Schematic depiction of the architecture and training procedure.

As the training criterion, we use the GAN Wasserstein loss [60] as

$$\Gamma(D,G) = \frac{1}{N}\sum_i D(y_i, x_i) - D(y_i, G(y_i, z_i)) \tag{1}$$

and we are interested in $\min_G \max_D \Gamma(D,G)$, meaning the parameters of G are optimized to minimize this loss, and the parameters of D are optimized to maximize it. Note that the optimization of G only affects the second term of Equation (1), resulting in a maximization of $D(y_i, G(y_i, z_i))$.

3.1.1. Architecture Details

For details on the implemented architecture, please refer to Table 1. We implement both the generator G and the critic D convolutional in time. This allows us to use less overlap (i.e., 50%) when chopping up the training data, as the convolutional architectures obtain differently shifted versions of the input by design. At test time, G is applied to variable-length and potentially relatively long input sequences (e.g., full songs). In such a setting, G does not perform very well if trained on short excerpts with zero-padding in the time dimension. Therefore, we do not use zero-padding for G during training.

The critic D is convolutional in time, too, resulting in as many loss outputs as there are spectrogram frames in the input (the individual costs are simply averaged for computing the final Wasserstein loss). We are using two convolutional groups throughout the critic stack, which amounts to two independent critics. Only in the output layer, those two groups are joined again. This is to provide D with two different views on the input, (1) the signed square-root of the complex STFT components (see Section 3.3.1) and (2) the *square-root* of the magnitude spectrum of the generator output (we found empirically that this resulted in more stable training than when using the log-magnitude spectrogram).

Many convolutional architectures with the same output and input size used for data restoration and in-painting employ the symmetrical U-Net paradigm (first introduced in [61]) with bottleneck layers and skip connections. In contrast, the architecture proposed in this work is non-symmetrical, mainly facilitating dilated convolutions for increasing receptive fields, and the main part of the architectures of G and D are identical (see Table 1). Only the top parts of the stacks differ, wherein in G the aggregated information is fed into deconvolution layers, while in D the information is used to compute the Wasserstein distance.

Table 1. Architecture details of generator G and critic D for four-second-long excerpts (i.e., 336 spectrogram frames), where $(\cdot)$-brackets mark information applying only to G, and information in $[\cdot]$-brackets applies only to D. During training, no padding is used in the time dimension for G, resulting in a shrinking of its output to 212 time-steps.

Layer	In Maps	Out Maps	Kernel Size	Dilation	Padding	Non-Linearity	Output Size
`Input`	-	-	-	-	-	-	$2 \times 1024 \times (336)[212]$
`Conv1`	2	18	3×3	1	$1, 1$	PReLU	$18 \times 1024 \times (336)[212]$
`Conv2`	18	38	3×3	2	$2, 2$	PReLU	$38 \times 1024 \times (336)[212]$
`Conv3`	38	38	3×3	4	$4, 4$	PReLU	$38 \times 1024 \times (336)[212]$
`Conv4`	38	4096	1024×1	1	$0, 0$	PReLU	$4096 \times 1 \times (336)[212]$
`Reshape1`		-	-	-	-	-	$128 \times 32 \times (336)[212]$
`ReMap`	128	256	1×1	1	$0, 0$	PReLU	$256 \times 32 \times (336)[212]$
`Conv5`	256	256	3×3	1	$1, (0)[1]$	PReLU	$256 \times 32 \times (334)[212]$
`(Noise Concat)`	-	-	-	-	-	-	$320 \times 32 \times 334$
`Conv6`	$(320)[256]$	256	3×3	2	$2, (0)[2]$	PReLU	$256 \times 32 \times (330)[212]$
`SelfGating`	-	-	-	-	-	-	$128 \times 32 \times (330)[212]$
`Conv7`	128	256	3×3	4	$4, (0)[4]$	PReLU	$256 \times 32 \times (322)[212]$
`SelfGating`	-	-	-	-	-	-	$128 \times 32 \times (322)[212]$
`Conv8`	128	256	3×3	8	$8, (0)[8]$	PReLU	$256 \times 32 \times (306)[212]$
`SelfGating`	-	-	-	-	-	-	$128 \times 32 \times (306)[212]$
`Conv9`	128	256	3×3	16	$16, (0)[16]$	PReLU	$256 \times 32 \times (274)[212]$
`SelfGating`	-	-	-	-	-	-	$128 \times 32 \times (274)[212]$
`Conv10`	128	256	3×3	1	$1, (0)[1]$	PReLU	$256 \times 32 \times (272)[212]$
`SelfGating`	-	-	-	-	-	-	$128 \times 32 \times (272)[212]$
`Conv11`	128	256	3×3	2	$2, (0)[2]$	PReLU	$256 \times 32 \times (268)[212]$
`SelfGating`	-	-	-	-	-	-	$128 \times 32 \times (268)[212]$
`Conv12`	128	256	3×3	4	$4, (0)[4]$	PReLU	$256 \times 32 \times (260)[212]$
`SelfGating`	-	-	-	-	-	-	$128 \times 32 \times (260)[212]$
`Conv13`	128	256	3×3	8	$8, (0)[8]$	PReLU	$256 \times 32 \times (244)[212]$
`SelfGating`	-	-	-	-	-	-	$128 \times 32 \times (244)[212]$
`Conv14`	128	256	3×3	16	$16, (0)[16]$	PReLU	$256 \times 32 \times (212)[212]$
`SelfGating`	-	-	-	-	-	-	$128 \times 32 \times 212$
`(Reshape2)`		-	-	-	-	-	$4096 \times 1 \times 212$
`(DeConv4)`	38	4096	1024×1	1	$0, 0$	PReLU	$38 \times 1024 \times 212$
`(DeConv3)`	38	38	3×3	4	$4, 4$	PReLU	$38 \times 1024 \times 212$
`(DeConv2)`	18	38	3×3	2	$2, 2$	PReLU	$18 \times 1024 \times 212$
`(DeConv1)`	2	18	3×3	1	$1, 1$	PReLU	$2 \times 1024 \times 212$
`(Output)`		-	-	-	-	-	$2 \times 1024 \times 212$
`[Conv15]`	128	256	3×3	1	$1, 1$	PReLU	$256 \times 32 \times 212$
`[Conv16]`	256	1	32×1	1	$0, 0$	-	$1 \times 1 \times 212$

We use Parametric Rectified Linear Units (PReLUs) [62] for all layers, and skip connections for D in the convolutional layers `Conv6`–`Conv14` in Table 1. The noise input z (to the generator G) is simply repeated in the two convolutional dimensions and concatenated to layer `Conv5`.

3.1.2. Gated Convolutions

In order to increase the architecture size under limited resources, a handy modification to common convolutions is self-gating convolutional layers. This idea was also proposed in [63], but we use PReLU activations instead of linear units for the gated output units (linear units resulted in unstable training). The characteristic of (self-)gating convolutions is that half the output maps of each convolutional layer is used to element-wise gate the other half of the output maps (where we use sigmoid non-linearities on the gating units and PReLUs on the gated units). We found that with self-gating layers, the network's performance does not degrade, even though the operation effectively halves the number

of output maps for each layer. The advantage of self-gating is a considerable reduction of memory, as the successive layer receives only half the input maps compared to non-self-gating layers. In practice, we use only one layer with twice as many output maps as input maps, which are then used for self-gating. Formally, we describe the operation with two different weight matrices as

$$y_l = \mathrm{PReLU}_l(x_l * W_l + a_l) \cdot \sigma(x_l * V_l + b_l) \tag{2}$$

where $x_l \in \mathbb{R}^{R \times P \times T}$ is the input to layer l with R convolutional input maps, $y_l \in \mathbb{R}^{S \times P \times T}$ is the resulting self-gated output of layer l with S convolutional output maps (where $S = R$ in our architecture), $W_l, V_l \in \mathbb{R}^{R \times S \times K \times K}$ are two weight matrices with quadratic kernels of size $K \times K$, and $a_l, b_l \in \mathbb{R}^S$ are bias vectors. PReLU are parametric ReLU activations [62], σ is the sigmoid non-linearity, $*$ is the convolution operator and $\cdot$ is the Hadamard product. This operation is applied to all convolutional layers `Conv6`-`Conv14` in Table 1.

3.1.3. Frequency Aggregation Filters

The neural network architectures used in audio processing are often derived from the visual domain. Convolutional neural networks are particularly well-suited for image processing because, in natural images, close pixels are usually more highly correlated than pixels which are further apart. Using stacks of convolutional layers, the filter kernels in the lower layers can learn highly correlated information, and filters in the higher layers can learn more complex combinations of filter responses of the lower layers. That way, as a rule of thumb, the higher up the information is in the convolutional hierarchy, the less correlated information is represented, which results in a hierarchical aggregation of pixel information that is well-suited for natural images.

Such a correlation assumption may also hold for the time dimension when working with musical audio data in spectrogram representation. However, it does not hold in the frequency dimension, where highly correlated spectral energy is potentially spread over the whole frequency range. In order to comply with this characteristic, it is common to employ non-rectangular filters in the input layer of a convolutional network stack, for example, kernels of shape [1, 32] in [64]. Nevertheless, when considering harmonics of tonal instruments or percussive sounds, correlated information may be so distant in the frequency axis that also with vertical filter kernels, a complete acoustic source may only be fully represented in the highest layer of the hierarchy. This fact contradicts the useful characteristic of efficient aggregation of information in convolutional network stacks to represent the least correlated information (i.e., the most complex patterns) in the highest layers of the hierarchy.

In order to tackle the problem of highly correlated frequency bins very distant in the frequency dimension, we take a novel approach. As it is not obvious before training which frequency bins are most correlated in the training data, and it is therefore not clear how to best design the architecture, we allow the network to *learn* a useful hierarchy of frequency aggregation during training (see Figure 2). To that end, in Layer `Conv4` (see Table 1), we use 4096 filter kernels that span the whole frequency dimension and only convolve in time (i.e., no padding in the frequency dimension). Then, we reshape the output maps (see `Reshape1` in Table 1) so that we again obtain a 2D convolutional architecture and let the network learn which filter kernels are most correlated, that is, in what layer of the hierarchy which filter responses need to be brought together (throughout layers `Conv5`–`Conv14`). In the generator G, `Reshape2` reshapes back to 4096 feature maps and `DeConv4` inverts the frequency aggregation.

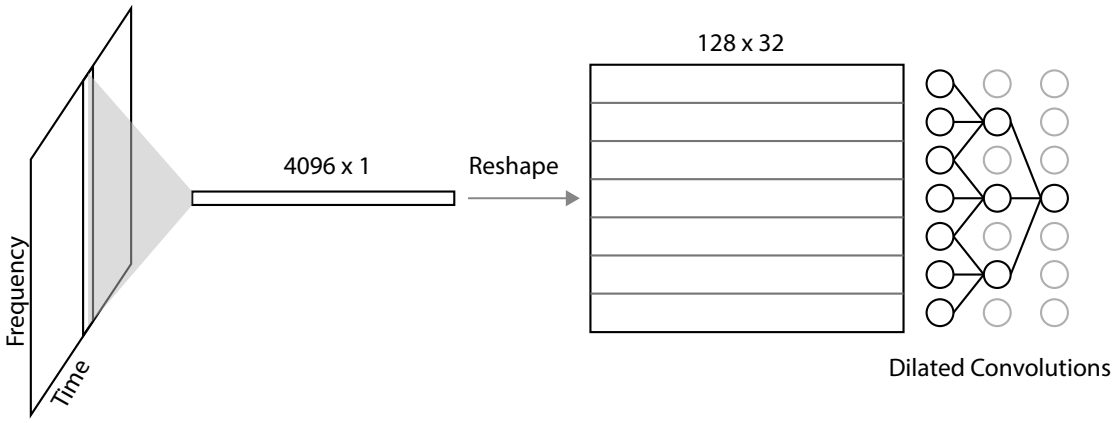


Figure 2. Schematic depiction of the Frequency Aggregation module for *one time-frame*. The frame is filtered with 4096 kernels resulting in filter responses of size 4096×1 (`Conv4` in Table 1). By reshaping, the responses are then separated into 32 groups (of size 128 each) and re-combined again through a stack of dilated convolutions (`Conv5`–`Conv14` in Table 1).

3.2. Training Procedure

Each model is trained for 40k iterations and a batch size of 12, which takes about 2 days on two NVIDIA Titan RTX with 24 GB memory each. We use the ADAM optimizer [65] with a learning rate of 1×10^{-3} and gradient penalty loss to restrict the gradients of D to 1 Lipschitz [66]. We also use a loss term that penalizes the magnitudes of the output of D for *real* input data, preventing the loss from drifting. Furthermore, He's initialization is performed for all the layers in the architecture [62].

3.2.1. Preventing Mode Collapse

We found that the Generator G tends to ignore the input noise z. This may be because G is densely conditioned on x, and the output variability in the data is limited given an input with a specific characteristic. In order to prevent such a mode collapse, to update G during training, we define an additional cost term which is maximal when the noise input to G does not influence the output of G. To that end, for a fixed y_i, we compute the ratio between the Euclidean distance of two arbitrary input noise vectors $\{z_i, z_j\}$, and the distances between the corresponding frequency profiles (summing over the time axis) and the rhythm profiles (summing over the frequency axis) of the output magnitude spectrogram of G, resulting in loss $\mathcal{L}_{\text{freq}}$ and $\mathcal{L}_{\text{rhyt}}$, respectively:

$$\mathcal{L}_{\text{profile}} = \frac{\vartheta \, \|z_i - z_j\|}{\left\| d^\intercal \hat{P}_{z_i}^{\circ \frac{1}{2}} - d^\intercal \hat{P}_{z_j}^{\circ \frac{1}{2}} \right\|_p^p} \tag{3}$$

where $\hat{P}_{z_i}^{\circ \frac{1}{2}}$ is the Hadamard root of the power spectrum of the output of G for input noise vector z_i, d is a column vector of 1s for the frequency profiles (resulting in the loss $\mathcal{L}_{\text{freq}}$) and a row vector of 1s for the rhythm profiles (resulting in $\mathcal{L}_{\text{rhyt}}$). The scalar ϑ controls the strength of the regularization, $p = 1.3$ for frequency profiles and $p = 1.6$ for rhythm profiles in our experiments.

In practice, for each conditional input y_i to G (i.e., each instance with index i in a batch), we compute two outputs $G(x_k, \{z_i, z_j\}_k)$ using randomly sampled $\{z_i, z_j\}_k \sim \mathcal{N}(0, I)$, and use those outputs to compute $\mathcal{L}_{\text{profile}}$, as well as the common gradient update of G. Note that in order to minimize $\mathcal{L}_{\text{profile}}$, G could simply learn to introduce huge changes in its output when the input noise z changes. However, in practice, this is prevented by the Wasserstein loss, which introduces a strong bias towards plausible (i.e., obeying the data distribution) outputs of G. Therefore, $\mathcal{L}_{\text{profile}}$ is effective in pushing the generations of G away from deterministic outputs while the overall training process remains stable.

3.3. Data

The model is trained on pairs of audio data, where one part is the MP3 version, and the other part is a high-quality (44.1 kHz) version of the signal. We use a dataset of approximately 64 h of Nr 1 songs of the U.S. charts between 1950 and 2020. The high-quality data is then compressed to 16 kbit/s, 32 kbit/s and 64 kbit/s mono MP3 using the LAME MP3 codec, version 3.100. (https://lame.sourceforge.io/ (accessed on 31 May 2021). The total number of songs is first divided into train, eval, and test sub-sets with a ratio of 80%, 10%, and 10%, respectively. We then split each of the songs into 4-s-long segments with 50% overlap for training and validation. For the subjective evaluation (see Section 3.4.5), we split the songs into segments of 8 seconds.

3.3.1. Data Representation

The main representation used in the proposed method are the complex STFT components of the audio data $h_{j,k} \in \mathbb{C}^{JK}$, as it has been shown that this representation works well for audio generation with GANs in [67]. The STFT is computed with a window size of 2048, and a hop size of 512. In addition, we perform non-linear scaling to all complex components, in order to obtain a scaling which is closer to human perception than when using the STFT components directly. That is, we transform each complex STFT coefficient $h_{j,k} = a_{j,k} + i\, b_{j,k}$ by taking the signed square-root of each of its components $h^{\sigma}_{j,k} = \sigma(a_{j,k}) + i\, \sigma(b_{j,k})$, where the signed square-root is defined as

$$\sigma(r) = \text{sign}(r)\sqrt{|r|} \tag{4}$$

3.4. Evaluation

We perform objective and subjective evaluations for the proposed method. The main goal of the evaluation is to assess the similarity between the reference signals (i.e., the high-quality signals) and the signal approximations (i.e., MP3 versions of the audio excerpts or outputs of the proposed model). The objective metrics used include Log-Spectral Distance (LSD), Mean Squared Error (MSE), Signal-to-Noise Ratio (SNR), Objective Difference Grade (ODG), and Distortion Index (DI). We also perform a subjective evaluation in the form of the Mean Opinion Score (MOS), which is described in Section 3.4.5.

3.4.1. Objective Difference Grade and Distortion Index

The Objective Difference Grade (ODG) is a computational approximation to subjective evaluations (i.e., the *subjective difference grade*) of users when comparing two signals. It ranges from 0 to -4, where lower values denote worse similarities between the signals. The Distortion Index (DI) is a metric that is differently scaled but correlated to the ODG and can be seen as the amount of distortion between two signals. Both the ODG and DI are based on a highly non-linear psychoacoustic model, including filtering and masking to approximate the human auditory perception. They are part of the Perceptual Evaluation of Audio Quality (PEAQ) ITU-R recommendation (BS.1387-1, last updated 2001) [68]. We use an openly available implementation of the basic version (as defined in the ITU recommendation) of PEAQ (https://github.com/akinori-ito/peaqb-fast (accessed on 31 May 2021), including the ODG and Distortion Index (DI). Even though PEAQ was initially designed for evaluating audio codecs with *minimal* coding artifacts, we found that the results correlated well with our perception.

3.4.2. Log-Spectral Distance

The log-spectral distance (LSD) is the Euclidean distance between the log-spectra of two signals and is invariant to phase information. Here, we calculate the LSD between

the spectrogram of the reference signal and that of the signal approximation. This results in the equation

$$LSD = \frac{1}{L}\sum_{l=0}^{L-1}\sqrt{\frac{1}{W}\sum_{f=0}^{W-1}\left[10\log_{10}\frac{P(l,f)}{\hat{P}(l,f)}\right]^2} \tag{5}$$

where P and $\hat{P}$ are the power spectra of x and $\hat{x}$, respectively, L is the total number of frames, and W is the total number of frequency bins.

3.4.3. Mean Squared Error

The LSD described above (see Section 3.4.2) is particularly high when comparing MP3 data with high-quality audio data. This is because it is standard practice found in many MP3 encoders (including the one we use) to perform a high-cut, removing most frequencies above a specific cut-off frequency. For values close to zero, log-scaling introduces negative numbers with very high magnitudes. Therefore, when comparing log-scaled power spectra of MP3 and PCM, we obtain particularly high distances. This generally favors algorithms that add frequencies in the upper range (like the proposed method). In this regard, a fairer comparison is the Mean Squared Error (MSE) between the *square-root* of the power spectra P of the two signals:

$$MSE = \frac{1}{L}\sum_{l=0}^{L-1}\frac{1}{W}\sum_{f=0}^{W-1}\left[\sqrt{P(l,f)}-\sqrt{\hat{P}(l,f)}\right]^2 \tag{6}$$

3.4.4. Signal-to-Noise Ratio

The signal-to-noise ratio (SNR) measures the ratio between a reference signal and the approximation residuals. As it is computed in the *time domain*, it is highly sensitive to phase information. The SNR is calculated as

$$\text{SNR} = 10\log_{10}\frac{\|s\|_2^2}{\|s-\hat{s}\|_2^2} \tag{7}$$

where s is the reference signal, and $\hat{s}$ is the signal approximation.

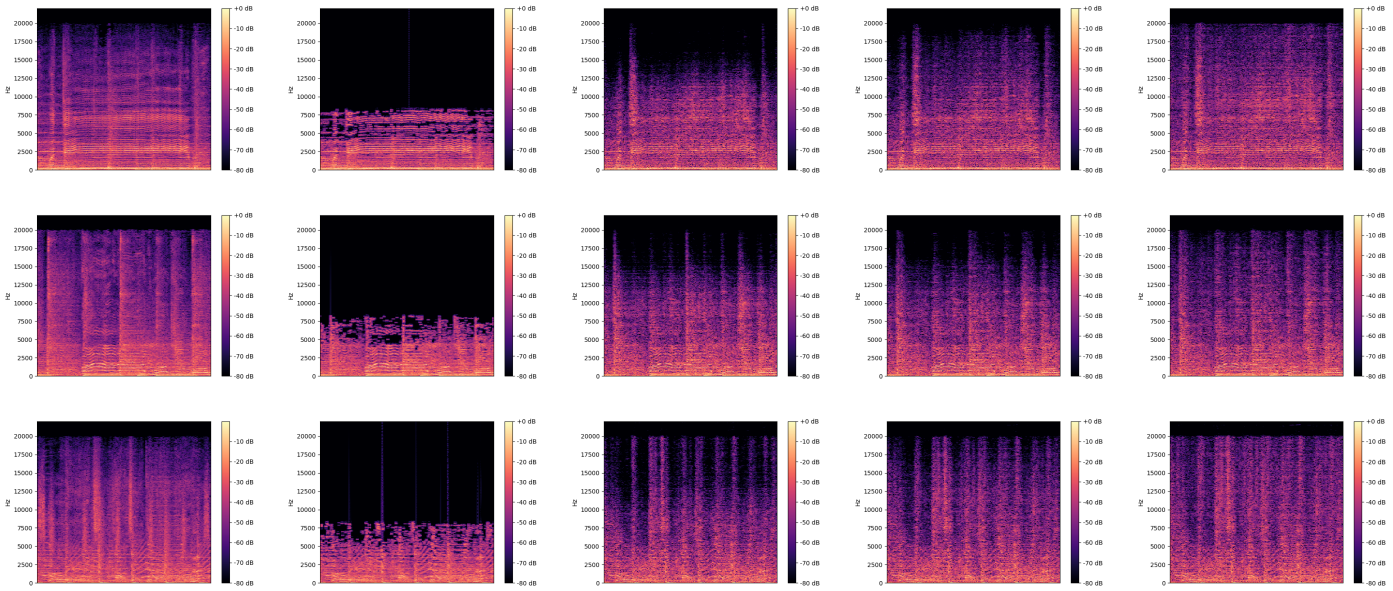
3.4.5. Mean Opinion Score

We asked 15 participants (mostly expert listeners) to provide absolute ratings (i.e., no reference audio excerpts) of the perceptual quality of isolated musical excerpts. The listening test was performed with random, eight-second-long audio excerpts of the test set. We presented to the listeners five high-quality audio excerpts, 15 MP3s (5×16 kbit/s, 5×32 kbit/s and 5×64 kbit/s) and 50 restored versions (using 25 stochastic restorations with random noise z and 25 deterministic restorations). Among these 25 restorations per model, we restored 10×16 kbit/s, 10×32 kbit/s and 5×64 kbit/s MP3s. All together, this results in 70 ratings per user.

The participants were asked to give an overall quality score and instructed to consider both the extent of the audible frequency range and noticeable, annoying artifacts. They provided their rating using a Likert-scale slider with five quality levels: (1) *very bad*, (2) *poor*, (3) *fair*, (4) *good* and (5) *excellent*). From these results, we computed the Mean Opinion Score (MOS) [69].

4. Results and Discussion

In the following, we present the results of the performed evaluations. In Section 4.1 we discuss the results of the objective metrics and in Section 4.2 we discuss the subjective evaluation (i.e., the MUSHRA test and the MOS). Figure 3 provides a visual impression of the model output by comparing the spectrograms of some high-quality audio segments, the corresponding MP3 versions, and some restorations.



(**a**) (**b**) (**c**) (**d**) (**e**)

Figure 3. Spectrograms of (**a**) original audio excerpts, (**b**) corresponding 32 kbit/s MP3 versions, and (**c**–**e**) restorations with different noise z randomly sampled from $\mathcal{N}(0, \mathbf{I})$.

4.1. Objective Evaluation

We tested the method for three different MP3 compression rates (16 kbit/s, 32 kbit/s and 64 kbit/s) as an input to the generator. Moreover, as stated above, we assumed that there were multiple valid solutions for an MP3 to be restored with very high compression rates. This would also mean that when using a stochastic generator, some of all possible samples should be closer to the original than when only using a deterministic generator. In order to test this hypothesis, for each compression rate, we train a stochastic generator (with noise input z) and a deterministic generator (without noise input). Then, for any input y taken from the test set, we sample 20 times with the corresponding generator using $z_i \sim \mathcal{N}(0, \mathbf{I})$, and for each objective metric, we take the best value of that set. Note that all objective metrics are computed by comparing the restored data with the original versions. Therefore, when picking samples to optimize a specific metric, we do not pick the sample with the best "quality", but rather the restoration that best approximates the original.

Table 2 and Figure 4 show the results (i.e., the comparison to the high-quality data) for the stochastic and the deterministic models, and the respective MP3 baselines. For high compression rates (i.e., 16 kbit/s and 32 kbit/s), the best reconstructions of the stochastic models generally perform better than the baseline MP3s in most metrics and improve over the outputs of the deterministic models. This indicates that the facilitation of a stochastic generator is actually useful for restoration tasks. For some metrics (except LSD), the deterministic models perform on a par with the MP3 baselines. That is reasonable, as there are many different ways to restore the original version, and it is unlikely that a deterministic model outputs a close approximation. In Figure 4 the strong violin-shaped forms in the figures indicate that the restorations form two groups in the ODG and DI metrics. From a visual inspection of the respective data, it becomes clear that those excerpts in the lower (worse) groups are without percussion instruments, indicating that the models cannot add meaningful high-frequency content for things such as singing voice or tonal instruments. The SNR is always worse for the restorations (compared to the MP3 baselines), which shows that the phase information is not faithfully regenerated. Given the high variety of possible phase information in the high-frequency range, particularly for percussive sounds, this is not surprising, but also does not hamper the perceived audio quality.

For the 64 kbit/s MP3s, we see that the reconstructions are worse than the MP3 itself, except in the LSD metric. Note that 64 kbit/s mono MP3s are already close to the original. The fact that the generator performs worse on these data indicates that in addition to adding high-frequency content (which is mostly advantageous, as can be seen in the LSD results), it also introduces some undesirable artifacts in the reconstruction of the MP3 information.

Table 2. Results of objective metrics for stochastic (`sto`) and deterministic (`det`) models and MP3 baselines (`mp3`) for different compression rates (16 kbit/s, 32 kbit/s, 64 kbit/s). Higher values are better for ODG, DI and SNR; lower values are better for LSD and MSE.

	ODG	DI	LSD	MSE	SNR
`mp3_16k`	−3.08	−1.67	10.98	0.40	**13.69**
`det_16k`	−3.12	−1.77	4.15	0.30	8.95
`sto_16k`	**−2.80**	**−1.19**	**3.72**	**0.26**	9.51
`mp3_32k`	−3.04	−1.56	9.75	0.31	**13.67**
`det_32k`	−2.99	−1.48	3.83	0.32	7.66
`sto_32k`	**−2.74**	**−1.07**	**3.75**	**0.26**	9.57
`mp3_64k`	**−2.64**	**−0.86**	4.89	**0.07**	**17.85**
`det_64k`	−2.95	−1.40	**3.54**	0.16	12.13
`sto_64k`	−2.74	−1.02	3.59	0.17	11.51

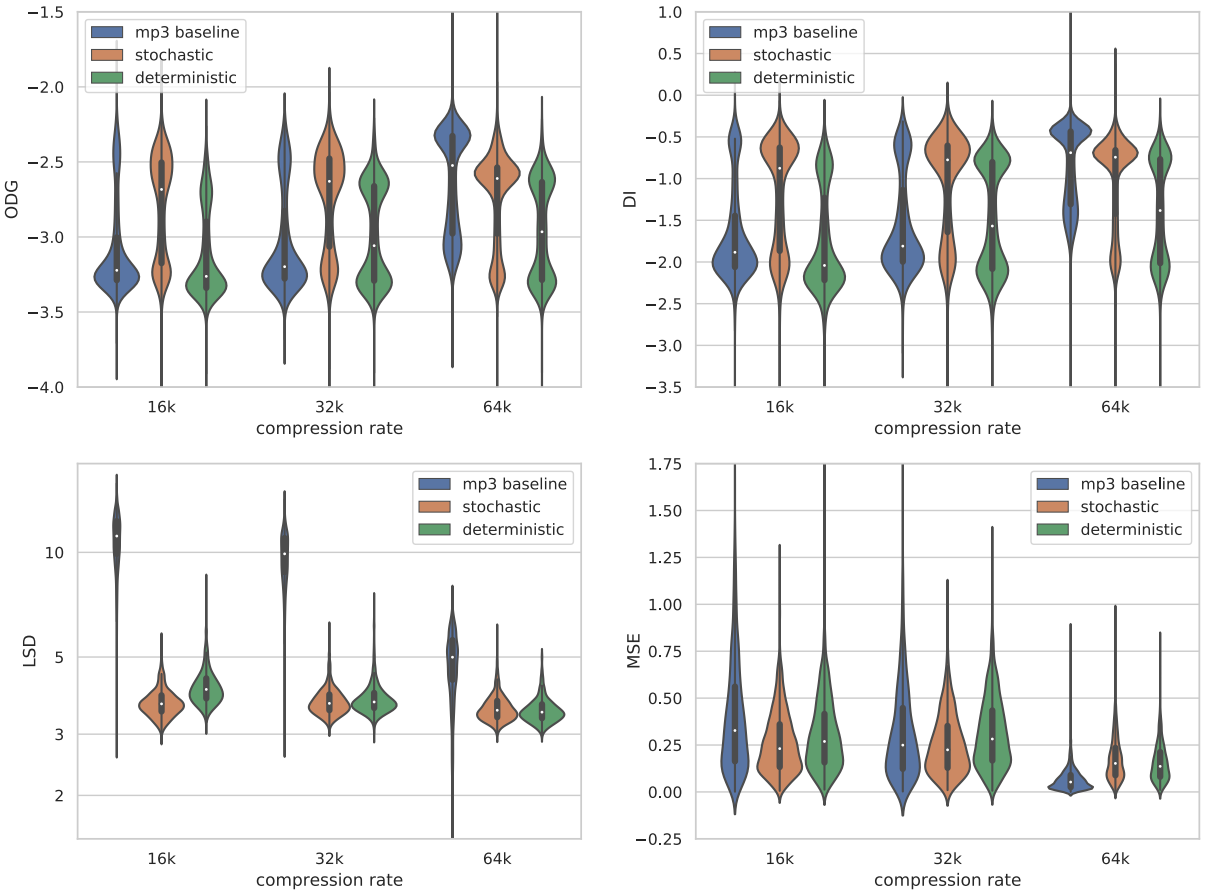


Figure 4. *Cont.*

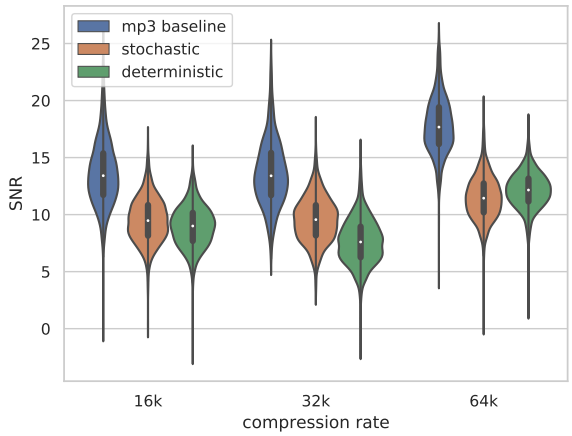


Figure 4. Violin plots of objective metrics for stochastic (`sto`) and deterministic (`det`) models and MP3 baselines (`mp3`) for different compression rates (16 kbit/s, 32 kbit/s, 64 kbit/s). Higher values are better for ODG, DI and SNR; lower values are better for LSD and MSE.

Frequency Profiles

In order to test the influence of the input noise z onto the generator output, we inputted random MP3 examples and restored them while keeping the noise input fixed. Then, we calculated the frequency profiles of the resulting outputs by taking the mean over the time dimension. Figure 5 shows examples of this experiment, which makes it clear that a specific z consistently causes a characteristic frequency profile over different examples. This is advantageous when z is chosen manually to control the restoration of an entire song, where a consistent characteristic is desired throughout the whole song.

4.2. Subjective Evaluation

In this section, we describe our own assessment when listening to the restored audio excerpts (Section 4.2.1), and then we provide results of the Mean Opinion Score (MOS), where we evaluate the restorations in a listening test with expert listeners.

4.2.1. Informal Listening

For sound examples of the proposed method, please refer to the accompanying website (available online: https://sonycslparis.github.io/restoration_mdpi_suppl_mat/ (accessed on 4 June 2021). When listening to the restored audio excerpts compared to the MP3 versions, the overall impression was a richer, higher bandwidth sound that could be described as "opening up". Additionally, we noticed that the model was able to remove some MP3 artifacts, particularly *swirlies*, as described in the introduction (see also [4]). It is clearly audible that the model adds frequency content which got lost in the MP3 compression. When comparing the restorations directly to the high-quality versions, it is noticeable that the level of detail in the high frequencies is considerably lower in the restorations. On closer inspection of the restorations, we could hear that for specific sound events, the model performed particularly well (i.e., adds convincing high-frequency content and removes specific compression artifacts), whereas other sources did not show considerable improvement, and some events tended to cause undesired, audible artifacts.

Among the sound events which generally improved very well are percussive elements like snare, crash, hi-hat, and cymbal sounds, but also other onsets with steep transients and non-harmonic high-frequency content, like the strumming of acoustic guitars or sibilants or plosives ('s' and 't') in a singing voice. Additionally, sustained electric guitars underwent considerable improvement. Note that all these sound types do not possess harmonics, but instead require the addition of high-frequency noise in the restoration process. Considering the nature of percussive sounds and the wide variety of sources in the training data, this is a reasonable outcome. On the one hand, percussive sounds dominate other sources in the higher frequency range, which constitutes the main difference between MP3 and high-

quality versions of the audio excerpts. On the other hand, harmonic sources are extremely varied, and their harmonics are of different characteristics. In addition, harmonics are rarely found above 10 kHz, which is the range in which the critic can best determine the difference between MP3 and high-quality audio signals.

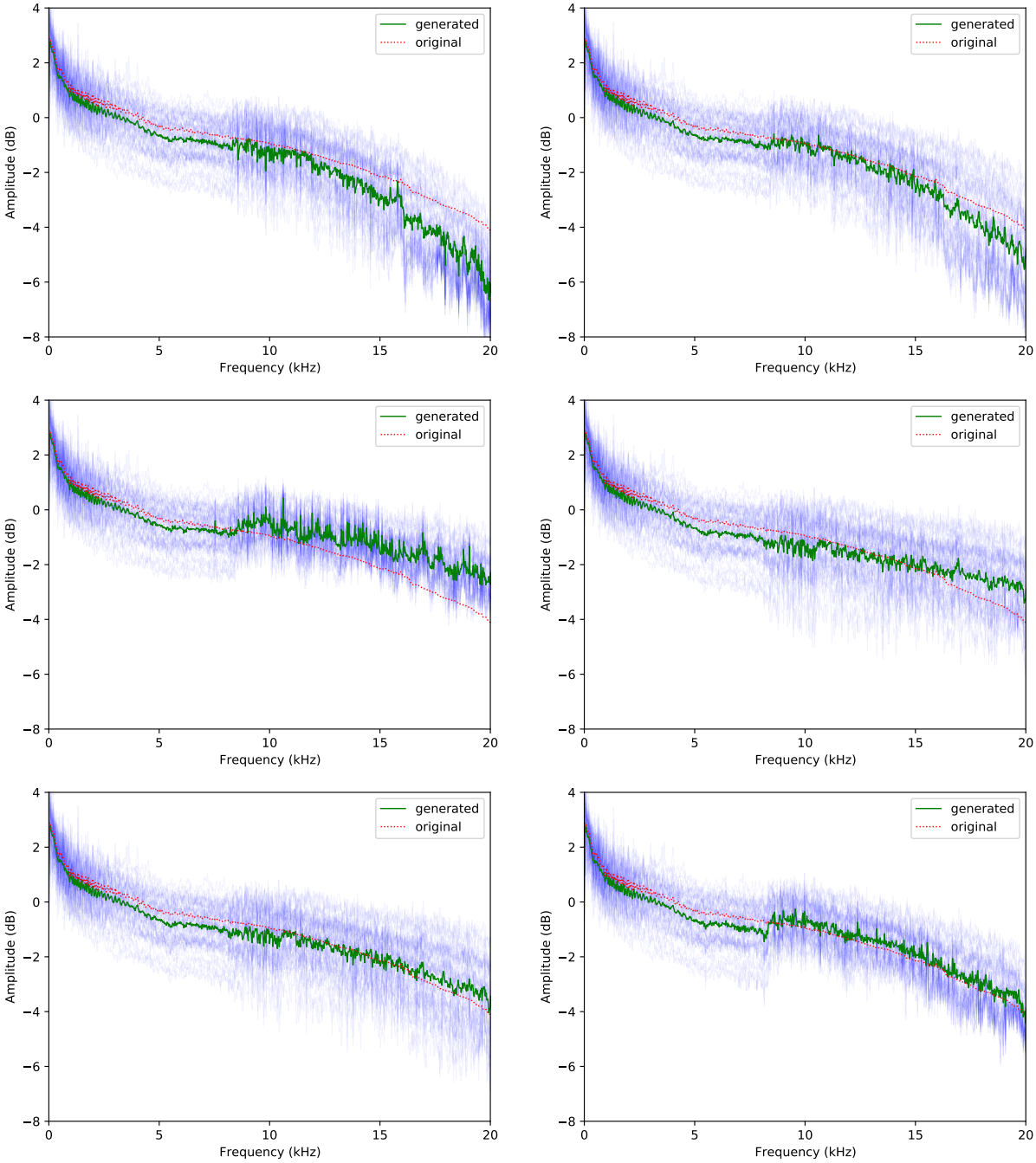


Figure 5. *Cont.*

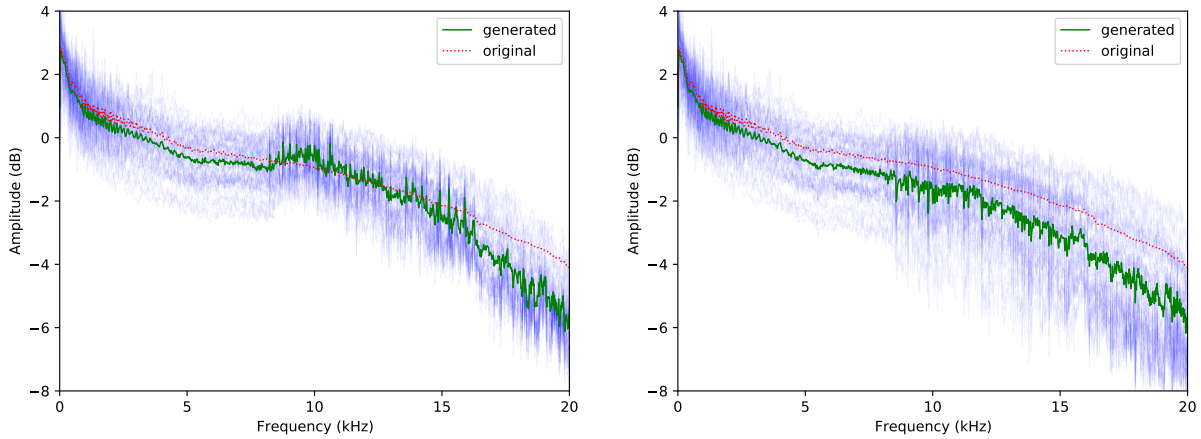


Figure 5. Frequency profiles of 50 random four-second-long excerpts from the test set (in 32 kbit/s) for different random input noise vectors z. The blue lines show the profiles of the individual samples, the green line shows the mean profile of the excerpts, and the dotted red line shows the mean of the high-quality excerpts for comparison. It becomes clear that z is strongly correlated with the energy in the upper bands and that a specific z yields a consistent overall characteristic.

Sometimes, the generator adds undesired, sustained noise, mainly when the audio input is very compressed or when there are rather loud, single tonal instruments or singing voices. Other undesired artifacts added by the generator are mainly "phantom percussions", like hi-hats that do not have meaningful rhythmic positions, triggered by events in the MP3 input that get confused with percussive sources. Additionally, the generator sometimes overemphasizes 's' or 't' phonemes of a singing voice. However, in some cases, percussive sounds not present in the original audio signals are added, which are rhythmically meaningful. In general, the overall characteristics of the percussion instruments are often different in the restorations compared to the high-quality versions. This is reasonable, as the lower frequencies present in the MP3 do not provide information about their characteristic in the higher frequency range, wherefore the characteristic needs to be regenerated by the model (dependent on the input noise z).

4.2.2. Formal Listening

Table 3 shows the results of the listening test (i.e., MOS ratings). Overall, the *original* and the 64 kbit/s MP3s (`mp3_64k`) obtained the highest ratings and the restored 64 kbit/s MP3s (`det_64k` and `sto_64k`) performed slightly worse. The ratings for the restored 16 kbit/s and 32 kbit/s (`det_16k`, `sto_16k`, `det_32k` and `sto_32k`) are considerably better than the MP3 versions (`mp3_16k` and `mp3_32k`). This shows that the proposed restoration process indeed results in better perceived audio quality. However, the random samples from the stochastic generators were not assessed as better than the outputs of the deterministic generators (the differences are not significant, as detailed below). We note that for the high compression rates, we reached only about half the average rating of the high-quality versions (but about double the rating of the MP3 versions). While overall, a restored MP3 version possesses a broader frequency range, weak ratings may result from off-putting artifacts, like the above-mentioned "phantom percussions". In eight-second-long excerpts, only one irritating artifact can already lead to a relatively weak rating for the whole example.

As the variance of the ratings is rather high, we also computed t-tests for statistical significance comparing responses to the different stimuli. We obtained p-values < 0.05 ($<10^{-5}$) when comparing `det` and `sto` to `mp3` for compression rates below 64 kbit/s. Conversely, we observed no statistically significant differences between ratings of `det` and `sto` for all compression rates (p-values > 0.15). Responses to *original* and `mp3_64k` also show no statistically significant differences (p-value $= 0.49$). We also observed no statistical

significance between responses to `mp3_64k` and `det_64k` (p-value $= 0.06$), whereas there is a significant difference between ratings of `sto_64k` and `mp3_64k` (p-value $= 0.04$).

Table 3. Mean Opinion Score (MOS) of absolute ratings for different compression rates. We compare the stochastic (*sto*) versions against the deterministic baselines (*det*), the MP3-encoded lower anchors (*mp3*) and the *original* high-quality audio excerpts.

	Mean	**std**
original	2.81	0.94
`mp3_16k`	0.74	0.79
`det_16k`	1.33	0.82
`sto_16k`	**1.40**	0.89
`mp3_32k`	0.80	0.71
`det_32k`	**1.43**	0.84
`sto_32k`	1.28	0.82
`mp3_64k`	**2.92**	0.95
`det_64k`	2.49	0.86
`sto_64k`	2.65	0.74

5. Conclusions and Future Work

We presented a Generative Adversarial Network (GAN) architecture for stochastic restoration of high-quality musical audio signals from highly compressed MP3 versions. We tested (1) whether the output of the proposed model improves the quality of the MP3 inputs, (2) whether the stochastic generator improves (i.e., can generate samples closer to the original) over a deterministic generator, and (3) whether the output of the stochastic variants are generally of higher quality than deterministic baseline models.

Results show that the restorations of the highly compressed MP3 versions (16 kbit/s and 32 kbit/s) are generally better than the MP3 versions themselves, which is reflected in a thorough objective evaluation, and confirmed in perceptual tests by human experts. We also tested weaker compression rates (64 kbit/s mono), where we found that the proposed architecture results in slightly worse results than the MP3 baseline. We could also show in the objective metrics that a stochastic generator can indeed output samples that are closer to the original than when using a deterministic generator. However, the perceptual tests indicate that when drawing random samples from the stochastic generator, the results are not assessed as significantly better than the results of the deterministic generator.

Due to the wide variety of popular music, the task of generating missing content is very challenging. However, the proposed models succeeded in adding high-frequency content for particular sources, resulting in an overall improved perceived quality of the music. Examples for sources where the model clearly learned to generate meaningful high-frequency content are percussive elements (i.e., snare, crash, hi-hat and cymbal sounds), sibilants or plosives ('s' and 't') in a singing voice, strummed acoustic guitars and (sustained) electric guitars.

We expect future improvements when limiting the style of the training data to particular genres or time periods of production. Additionally, as we use the complex spectrum directly, the adaption to Complex Networks [58] could improve the results further. In order to tackle the problem of "phantom percussions" (as described in Section 4.2.1), a beat detection algorithm could provide additional information to the generator so that it is better informed about the rhythmic structure of the input. For improvement in learning to restore the harmonics of tonal sources, other representations (e.g., Magnitude and Instantaneous Frequencies (Mag-IF) [70]) or a different scaling (e.g., Mel-scaled spectrograms) could be tested for the input and output of the generator.

6. Author Biography

Stefan Lattner is an associate researcher at Sony CSL Paris. He earned his Ph.D. degree in 2019 at the JKU Linz, at the CP Institute, under the supervision of Gerhard Widmer. He received his MSc. degree (in Informatics) in 2013 at the JKU Linz, and his BSc. degree (2009) in Mediatechnology and -design at the University of Applied Sciences in Hagenberg, Upper Austria. From 2013 to 2018, Lattner worked at the Austrian Research Institute for Artificial Intelligence in Vienna, and between 2009 and 2014, he was lead developer and project manager at Re-Compose. Lattner is concerned with musical structure learning, invariance learning, music and audio generation, as well as meta-learning and information theory in music cognition. He received the Best Paper Award for the paper "Learning Complex Basis Functions for Invariant Representations of Audio" at the ISMIR conference, Delft, 2019.

Javier Nistal is an assistant researcher at Sony CSL Paris and Ph.D. candidate at Telecom IP Paris under a Marie Curie fellowship. His current research is centered around controllable audio synthesis with Generative Adversarial Networks. Before starting his Ph.D., Nistal was a Machine Learning Researcher at MIDAS, a mixing console manufacturing company, which is part of Music Tribe. During his time at MIDAS, Nistal worked on instrument recognition and spill detection for the MIDAS Heritage D, the first mixing console which integrates Machine Learning. Before joining Music Tribe, Nistal interned in Jukedeck and SoundCloud as an MIR researcher. Nistal earned an MSc in Sound and Music Computing from the Music Technology Group (MTG) at Pompeu Fabra University and a BSc degree in Sound and Image Engineering from the Polytechnic University of Madrid.

Author Contributions: Conceptualization, S.L.; methodology, S.L.; implementation, S.L., J.N.; obj. validation, S.L.; subj. validation, J.N., S.L.; investigation, S.L.; data curation, S.L.; writing—original draft preparation, S.L., J.N.; writing—review and editing, S.L.; visualization, S.L.; supervision, S.L.; project administration, S.L. Both authors have read and agreed to the published version of the manuscript.

Funding: This research has received founding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 765068

Institutional Review Board Statement: Ethical review and approval were waived for this study, as the formal listening experiment was performed voluntary, anonymous, and no unsettling material was presented to the subjects.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Generated audio examples produced in this study can be found at https://sonycslparis.github.io/restoration_mdpi_suppl_mat/ (accessed on 4 June 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Brandenburg, K.; Stoll, G. ISO/MPEG-1 audio: A generic standard for coding of high-quality digital audio. *J. Audio Eng. Soc.* **1994**, *42*, 780–792.
2. Brandenburg, K. MP3 and AAC explained. In Proceedings of the Audio Engineering Society Conference: 17th International Conference: High-Quality Audio Coding, Audio Engineering Society, Florence, Italy, 2–5 September 1999.
3. Musmann, H.G. Genesis of the MP3 audio coding standard. *IEEE Trans. Consum. Electron.* **2006**, *52*, 1043–1049. [CrossRef]
4. Corbett, I. What Data Compression Does to Your Music. Available online: https://www.soundonsound.com/techniques/what-data-compression-does-your-music (accessed on 31 May 2021).
5. Williamson, D.S.; Wang, D. Speech dereverberation and denoising using complex ratio masks. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, 5–9 March 2017; pp. 5590–5594. [CrossRef]
6. Kumar, R.; Kumar, K.; Anand, V.; Bengio, Y.; Courville, A.C. NU-GAN: High resolution neural upsampling with GAN. *arXiv* **2020**, arXiv:2010.11362.
7. Zhao, Z.; Liu, H.; Fingscheidt, T. Convolutional Neural Networks to Enhance Coded Speech. *IEEE ACM Trans. Audio Speech Lang. Process.* **2019**, *27*, 663–678. [CrossRef]

8. Fisher, K.; Scherlis, A. WaveMedic: Convolutional Neural Networks for Speech Audio Enhancement. 2016. Available online: http://cs229.stanford.edu/proj2016/report/FisherScherlis-WaveMedic-project.pdf (accessed on 4 June 2021).
9. Skoglund, J.; Valin, J. Improving Opus Low Bit Rate Quality with Neural Speech Synthesis. In Proceedings of the Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25–29 October 2020; Meng, H., Xu, B., Zheng, T.F., Eds.; ISCA: Baixas, France, 2020; pp. 2847–2851. [CrossRef]
10. Biswas, A.; Jia, D. Audio Codec Enhancement with Generative Adversarial Networks. In Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, 4–8 May 2020; pp. 356–360. [CrossRef]
11. Porov, A.; Oh, E.; Choo, K.; Sung, H.; Jeong, J.; Osipov, K.; Francois, H. Music enhancement by a novel CNN architecture. In *Audio Engineering Society Convention 145*; Audio Engineering Society: New York, NY, USA, 2018.
12. Park, S.R.; Lee, J. A Fully Convolutional Neural Network for Speech Enhancement. In Proceedings of the Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, 20–24 August 2017; Lacerda, F., Ed.; ISCA: Baixas, France, 2017; pp. 1993–1997.
13. Gupta, A.; Shillingford, B.; Assael, Y.M.; Walters, T.C. Speech Bandwidth Extension with Wavenet. In Proceedings of the 2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, WASPAA 2019, New Paltz, NY, USA, 20–23 October 2019; pp. 205–208. [CrossRef]
14. Isik, U.; Giri, R.; Phansalkar, N.; Valin, J.; Helwani, K.; Krishnaswamy, A. PoCoNet: Better Speech Enhancement with Frequency-Positional Embeddings, Semi-Supervised Conversational Data, and Biased Loss. In Proceedings of the Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25–29 October 2020; Meng, H., Xu, B., Zheng, T.F., Eds.; ISCA: Baixas, France, 2020; pp. 2487–2491. [CrossRef]
15. Hu, Y.; Liu, Y.; Lv, S.; Xing, M.; Zhang, S.; Fu, Y.; Wu, J.; Zhang, B.; Xie, L. DCCRN: Deep Complex Convolution Recurrent Network for Phase-Aware Speech Enhancement. In Proceedings of the Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25–29 October 2020; Meng, H., Xu, B., Zheng, T.F., Eds.; 2020, pp. 2472–2476. [CrossRef]
16. Kontio, J.; Laaksonen, L.; Alku, P. Neural Network-Based Artificial Bandwidth Expansion of Speech. *IEEE Trans. Speech Audio Process.* **2007**, *15*, 873–881. [CrossRef]
17. Li, K.; Lee, C. A deep neural network approach to speech bandwidth expansion. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, QLD, Australia, 19–24 April 2015; pp. 4395–4399. [CrossRef]
18. Xu, Y.; Du, J.; Dai, L.; Lee, C. A Regression Approach to Speech Enhancement Based on Deep Neural Networks. *IEEE ACM Trans. Audio Speech Lang. Process.* **2015**, *23*, 7–19. [CrossRef]
19. Lagrange, M.; Gontier, F. Bandwidth Extension of Musical Audio Signals with No Side Information Using Dilated Convolutional Neural Networks. In Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, 4–8 May 2020; pp. 801–805. [CrossRef]
20. Miron, M.; Davies, M. High frequency magnitude spectrogram reconstruction for music mixtures using convolutional autoencoders. In Proceedings of the 21st Int. Conference on Digital Audio Effects (DAFx-18), Aveiro, Portugal, 4–8 September 2018; pp. 173–180.
21. Deng, J.; Schuller, B.W.; Eyben, F.; Schuller, D.; Zhang, Z.; Francois, H.; Oh, E. Exploiting time-frequency patterns with LSTM-RNNs for low-bitrate audio restoration. *Neural Comput. Appl.* **2020**, *32*, 1095–1107. [CrossRef]
22. Maiti, S.; Mandel, M.I. Parametric Resynthesis with Neural Vocoders. In Proceedings of the 2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, WASPAA 2019, New Paltz, NY, USA, 20–23 October 2019; pp. 303–307. [CrossRef]
23. Dhariwal, P.; Jun, H.; Payne, C.; Kim, J.W.; Radford, A.; Sutskever, I. Jukebox: A Generative Model for Music. *arXiv* **2020**, arXiv:2005.00341.
24. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.C.; Bengio, Y. Generative Adversarial Nets. In Proceedings of the NIPS, Montréal, QC, Canada, 8–13 December 2014.
25. Nistal, J.; Lattner, S.; Richard, G. DrumGAN: Synthesis of Drum Sounds with Timbral Feature Conditioning Using Generative Adversarial Networks. In Proceedings of the 21st International Society for Music Information Retrieval, ISMIR, Virtual Conference, Virtual, 11–16 October 2020.
26. Larsen, E.; Aarts, R.M. *Audio Bandwidth Extension: Application of Psychoacoustics, Signal Processing and Loudspeaker Design*; John Wiley & Sons: Hoboken, NJ, USA, 2005.
27. Bansal, D.; Raj, B.; Smaragdis, P. Bandwidth expansion of narrowband speech using non-negative matrix factorization. In Proceedings of the INTERSPEECH 2005—Eurospeech, 9th European Conference on Speech Communication and Technology, Lisbon, Portugal, 4–8 September 2005; pp. 1505–1508.
28. Makhoul, J.; Berouti, M.G. High-frequency regeneration in speech coding systems. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '79, Washington, DC, USA, 2–4 April 1979; pp. 428–431. [CrossRef]
29. Dietz, M.; Liljeryd, L.; Kjorling, K.; Kunz, O. Spectral Band Replication, a novel approach in audio coding. In *Audio Engineering Society Convention 112*; Audio Engineering Society: New York, NY, USA, 2002.

30. Mandel, M.I.; Cho, Y.S. Audio super-resolution using concatenative resynthesis. In Proceedings of the 2015 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, WASPAA 2015, New Paltz, NY, USA, 18–21 October 2015; pp. 1–5. [CrossRef]
31. Dong, J.; Wang, W.; Chambers, J.A. Audio super-resolution using analysis dictionary learning. In Proceedings of the 2015 IEEE International Conference on Digital Signal Processing, DSP 2015, Singapore, 21–24 July 2015; pp. 604–608. [CrossRef]
32. Dong, C.; Loy, C.C.; He, K.; Tang, X. Image Super-Resolution Using Deep Convolutional Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 295–307. [CrossRef]
33. Isola, P.; Zhu, J.; Zhou, T.; Efros, A.A. Image-to-Image Translation with Conditional Adversarial Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017; IEEE Computer Society: Los Alamitos, CA, USA, 2017; pp. 5967–5976. [CrossRef]
34. Kuleshov, V.; Enam, S.Z.; Ermon, S. Audio Super-Resolution using Neural Networks. In Proceedings of the 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017.
35. Lim, T.; Yeh, R.A.; Xu, Y.; Do, M.N.; Hasegawa-Johnson, M. Time-Frequency Networks for Audio Super-Resolution. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, 15–20 April 2018; pp. 646–650. [CrossRef]
36. Ling, Z.; Ai, Y.; Gu, Y.; Dai, L. Waveform Modeling and Generation Using Hierarchical Recurrent Neural Networks for Speech Bandwidth Extension. *IEEE ACM Trans. Audio Speech Lang. Process.* **2018**, *26*, 883–894. [CrossRef]
37. Loizou, P. *Speech Enhancement: Theory and Practice*; CRC Press: Boca Raton, FL, USA, 2007. [CrossRef]
38. Ortega-Garcia, J.; Gonzalez-Rodriguez, J. Overview of speech enhancement techniques for automatic speaker recognition. In Proceedings of the 4th International Conference on Spoken Language Processing, Philadelphia, PA, USA, 3–6 October 1996.
39. Seltzer, M.L.; Yu, D.; Wang, Y. An investigation of deep neural networks for noise robust speech recognition. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, 26–31 May 2013; pp. 7398–7402. [CrossRef]
40. Kolbæk, M.; Tan, Z.; Jensen, J. Speech enhancement using Long Short-Term Memory based recurrent Neural Networks for noise robust Speaker Verification. In Proceedings of the 2016 IEEE Spoken Language Technology Workshop, SLT 2016, San Diego, CA, USA, 13–16 December 2016; pp. 305–311. [CrossRef]
41. Yang, L.P.; Fu, Q.J. Spectral subtraction-based speech enhancement for cochlear implant patients in background noise (L). *J. Acoust. Soc. Am.* **2005**, *117*, 1001–1004. [CrossRef]
42. Chen, J.; Wang, Y.; Yoho, S.; Wang, D.; Healy, E. Large-scale training to increase speech intelligibility for hearing-impaired listeners in novel noises. *J. Acoust. Soc. Am.* **2016**, *139*, 2604–2612. [CrossRef]
43. Boll, S. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Trans. Acoust. Speech, Signal Process.* **1979**, *27*, 113–120. [CrossRef]
44. Lim, J.; Oppenheim, A. All-pole modeling of degraded speech. *IEEE Trans. Acoust. Speech Signal Process.* **1978**, *26*, 197–210. [CrossRef]
45. Ephraim, Y. Statistical-model-based speech enhancement systems. *Proc. IEEE* **1992**, *80*, 1526–1555. [CrossRef]
46. Dendrinos, M.; Bakamidis, S.; Carayannis, G. Speech enhancement from noise: A regenerative approach. *Speech Commun.* **1991**, *10*, 45–57. [CrossRef]
47. Williamson, D.S.; Wang, Y.; Wang, D. Complex Ratio Masking for Monaural Speech Separation. *IEEE ACM Trans. Audio Speech Lang. Process.* **2016**, *24*, 483–492. [CrossRef] [PubMed]
48. Erdogan, H.; Hershey, J.R.; Watanabe, S.; Roux, J.L. Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, QLD, Australia, 19–24 April 2015; pp. 708–712. [CrossRef]
49. Pascual, S.; Bonafonte, A.; Serrà, J. SEGAN: Speech Enhancement Generative Adversarial Network. In Proceedings of the Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, 20–24 August 2017; Lacerda, F., Ed.; ISCA: Baixas, France, 2017; pp. 3642–3646.
50. Pascual, S.; Serrà, J.; Bonafonte, A. Towards Generalized Speech Enhancement with Generative Adversarial Networks. In Proceedings of the Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15–19 September 2019; Kubin, G., Kacic, Z., Eds.; 2019; pp. 1791–1795. [CrossRef]
51. Li, Z.; Dai, L.; Song, Y.; McLoughlin, I.V. A Conditional Generative Model for Speech Enhancement. *Circuits Syst. Signal Process.* **2018**, *37*, 5005–5022. [CrossRef]
52. Donahue, C.; Li, B.; Prabhavalkar, R. Exploring Speech Enhancement with Generative Adversarial Networks for Robust Speech Recognition. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, 15–20 April 2018; pp. 5024–5028. [CrossRef]
53. Michelsanti, D.; Tan, Z. Conditional Generative Adversarial Networks for Speech Enhancement and Noise-Robust Speaker Verification. In Proceedings of the Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, 20–24 August 2017; Lacerda, F., Ed.; 2017; pp. 2008–2012.

54. Fu, S.; Liao, C.; Tsao, Y.; Lin, S. MetricGAN: Generative Adversarial Networks based Black-box Metric Scores Optimization for Speech Enhancement. In Proceedings of the 36th International Conference on Machine Learning, ICML 2019, Long Beach, CA, USA, 9–15 June 2019; Proceedings of Machine Learning Research; Chaudhuri, K., Salakhutdinov, R., Eds.; Volume 97, pp. 2031–2041.
55. Phan, H.; McLoughlin, I.V.; Pham, L.D.; Chén, O.Y.; Koch, P.; Vos, M.D.; Mertins, A. Improving GANs for Speech Enhancement. *IEEE Signal Process. Lett.* **2020**, *27*, 1700–1704. [CrossRef]
56. Germain, F.G.; Chen, Q.; Koltun, V. Speech Denoising with Deep Feature Losses. In Proceedings of the Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15–19 September 2019; Kubin, G., Kacic, Z., Eds.; 2019; pp. 2723–2727. [CrossRef]
57. Su, J.; Jin, Z.; Finkelstein, A. HiFi-GAN: High-Fidelity Denoising and Dereverberation Based on Speech Deep Features in Adversarial Networks. In Proceedings of the Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25–29 October 2020; Meng, H., Xu, B., Zheng, T.F., Eds.; 2020; pp. 4506–4510. [CrossRef]
58. Trabelsi, C.; Bilaniuk, O.; Zhang, Y.; Serdyuk, D.; Subramanian, S.; Santos, J.F.; Mehri, S.; Rostamzadeh, N.; Bengio, Y.; Pal, C.J. Deep Complex Networks. In Proceedings of the 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, 30 April–3 May 2018.
59. Zhao, S.; Nguyen, T.H.; Ma, B. Monaural Speech Enhancement with Complex Convolutional Block Attention Module and Joint Time Frequency Losses. *arXiv* **2021**, arXiv:2102.01993.
60. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein Generative Adversarial Networks. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; Proceedings of Machine Learning Research; Precup, D., Teh, Y.W., Eds.; Volume 70, pp. 214–223.
61. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, 7–12 June 2015; IEEE Computer Society: Los Alamitos, CA, USA, 2015; pp. 3431–3440. [CrossRef]
62. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In Proceedings of the 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, 7–13 December 2015; pp. 1026–1034. [CrossRef]
63. Dauphin, Y.N.; Fan, A.; Auli, M.; Grangier, D. Language Modeling with Gated Convolutional Networks. In Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6–11 August 2017; Precup, D., Teh, Y.W., Eds.; 2017; Volume 70, pp. 933–941.
64. Pons Puig, J. Deep Neural Networks for Music and Audio Tagging. Ph.D. Thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2019.
65. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015.
66. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A.C. Improved Training of Wasserstein GANs. In Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017; pp. 5767–5777.
67. Nistal, J.; Lattner, S.; Richard, G. Comparing Representations for Audio Synthesis Using Generative Adversarial Networks. In Proceedings of the 28th European Signal Processing Conference, EUSIPCO 2020, Amsterdam, The Netherlands, 18–21 January 2021; pp. 161–165. [CrossRef]
68. Thiede, T.; Treurniet, W.C.; Bitto, R.; Schmidmer, C.; Sporer, T.; Beerends, J.G.; Colomes, C. PEAQ-The ITU standard for objective measurement of perceived audio quality. *J. Audio Eng. Soc.* **2000**, *48*, 3–29.
69. *ITU-T Recommendation: Vocabulary for Performance and Quality of Service*; International Telecommunications Union—Radiocommunication (ITU-T), R.I.T.P.: Geneva, Switzerland, 2006.
70. Engel, J.H.; Agrawal, K.K.; Chen, S.; Gulrajani, I.; Donahue, C.; Roberts, A. GANSynth: Adversarial Neural Audio Synthesis. In Proceedings of the 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, 6–9 May 2019.