LEARNING TO DENOISE HISTORICAL MUSIC

Yunpeng Li **Beat Gfeller** **Marco Tagliasacchi** **Dominik Roblek**
Google
{yunpeng,beatg,mtagliasacchi,droblek}@google.com

ABSTRACT

We propose an audio-to-audio neural network model that learns to denoise old music recordings. Our model internally converts its input into a time-frequency representation by means of a short-time Fourier transform (STFT), and processes the resulting complex spectrogram using a convolutional neural network. The network is trained with both reconstruction and adversarial objectives on a synthetic noisy music dataset, which is created by mixing clean music with real noise samples extracted from quiet segments of old recordings. We evaluate our method quantitatively on held-out test examples of the synthetic dataset, and qualitatively by human rating on samples of actual historical recordings. Our results show that the proposed method is effective in removing noise, while preserving the quality and details of the original music.

1. INTRODUCTION

Archives of historical music recordings are an important means for preserving cultural heritage. Most such records, however, were created with outdated equipment, and stored on analog media such as phonograph records and wax cylinders. The technological limitation of the recording process and the subsequent deterioration of the storage media inevitably left their marks, manifested by the characteristic crackling, clicking, and hissing noises that are typical in old records. While "remastering" employed by the recording industry can substantially improve the sound quality, it is a time-consuming process of manual labor. The focus of this paper is an automated method that learns from data to remove noise and restore music.

Audio denoising has a long history in signal processing [1]. Traditional methods typically use a simplified statistical model of the noise, whose parameters are estimated from the noisy audio. Examples of these techniques are spectral noise subtraction [2, 3], spectral masking [4, 5], statistical methods based on Wiener filtering [6] and Bayesian estimators [7, 8]. Many of these approaches, however, focus on speech. Moreover, they often make simplifying assumptions about the structure of the noise, which makes them less effective on non-stationary real-world noise.

Recent advances in deep learning saw the emergence of data-driven methods that do not make such *a priori* assumptions about noise. Instead they learn an implicit noise model from training examples, which typically consist of pairs of clean and noisy versions of the same audio in a supervised setup. Crucial challenges facing the adoption of the deep learning paradigm for our task are: i) can we design a model powerful enough for the complexity of music, yet simple and fast enough to be practical, and ii) how can we train such a model, given that we have no clean ground truth for historical recordings? In this paper, we address these issues and show that it is indeed feasible to build an effective and efficient model for music denoising.

1.1 Related Work

Sparse linear regression with structured priors is used in [9] to denoise music from synthetically added white Gaussian noise, obtaining large SNR improvements on a "glockenspiel" excerpt, and on an Indian polyphonic song. [10] considers the problem of removing artifacts of perceptual coding audio compression with low bit-rates. That work, which uses LSTMs, is the first successful application of deep learning for this type of music audio restoration. Note that in contrast to our work, aligned pairs of original and compressed audio samples are readily available. Statistical methods are applied in [11] to denoise Greek Folk music recorded in outdoor festivities. In [12], the author applies structured sparsity models to two specific audio recordings that were digitized from wax cylinders, and describes the results qualitatively. In [13], the authors describe how to fill in gaps (at known positions) of several seconds in music audio, using self-similar parts from the recording itself.

Our method is also related to audio super-resolution, also known as bandwidth extension. This is the process of extending audio from low to higher sample rates, which requires restoring the high frequency content. In [14, 15] two approaches which work for music are described. On piano music, for example, [15] obtains an SNR of 19.3 when upsampling a low-pass filtered audio from 4kHz to 16kHz.

Many existing denoising approaches focus on speech instead of music [16–19]. Given that these two domains have very different properties, it is not clear a priori how well such methods transfer to the music domain. Nevertheless, our work is inspired by recent approaches that use generative adversarial networks (GANs) to improve the quality of audio [18, 20, 21]. For example, [21] obtains

Attribution: Y. Li, B. Gfeller, M. Tagliasacchi, and D. Roblek, "Learning to Denoise Historical Music", in *Proc. of the 21st Int. Society for Music Information Retrieval Conf.,* Montréal, Canada, 2020.

significant improvements denoising speech and applause sounds that have been decoded at a low bit-rate, using a wave-to-wave convolutional architecture.

In this paper, we present a method to remove noise from historical music recordings, using two sources of audio: i) a collection of historical music recordings to be restored, for which no clean reference is available, and ii) a separate collection of music of the same genre that contains high-quality recordings. We focus on classical music, for which both public domain historical recordings as well as modern digital recordings are available. This paper makes the following contributions:

- We provide a fully automated approach that succeeds in removing noise from historical recordings, while preserving the musical content in high quality. Quality is measured in terms of SNR and subjective scores inspired by MUSHRA [22], and examples on real historical recordings are provided [1] .
- Our approach employs a new architecture that transforms audio in the time domain, using a multi-scale approach, combined with STFT and inverse STFT. As this architecture is able to output high-quality music, it may be a useful architecture for other tasks that involve the transformation of music audio.
- We provide an efficient and fully automated method to extract noise segments (without music) from a collection of historical music recordings. This is a key ingredient of our approach, as it allows us to create synthetic pairs of <clean, noisy> audio samples.

The rest of this paper is organized as follows. Our approach is described in detail in Section 2, and experimental results are given in Section 3. We conclude in Section 4.

2. METHOD

Our model is an audio-to-audio generator learned from paired examples with both reconstruction and adversarial objectives.

2.1 Creating paired training examples

For training, we use time-aligned pairs of <clean, noisy> examples, where clean music is used as targets, and noisy music as inputs to the generator. We take a data-driven approach to generate noisy audio from clean references. We synthesize noisy samples by simulating the degradation process affecting the historical recordings, namely applying band-pass filtering, followed by additive mixing with noise samples extracted from "quasi-silence" segments of historical recordings.

Specifically, we scan the noisy historical recordings looking for low-energy segments in the time domain, which corresponds to pauses in the musical scores. To this end, we compute the rolling standard deviation from the raw audio samples with a window size equal to 100ms. Then, we estimate an adaptive threshold τ based on the q-th quantile of the standard deviations and keep the segments that satisfy the following two conditions: i) the local standard deviation is below τ, and ii) the segment has a minimum duration of T. Intuitively, the value of q is selected based on a trade-off between the number of extracted segments and the need of extracting noise-only segments. In our experiments, we set $q = 0.5\%$ and $T = 100$ms. In this way, from 801 different recordings, we are able to extract around 8900 noise samples.

From each of these short noise segments, we need to generate noise samples having the same length as the clean audio references. We do this by replicating the noise segment in time, using overlap-and-add (OLA) with an overlap equal to 20% of the segment length. Given the short duration of most noise segments, this operation alone would lead to periodic noise patterns which differ from the noise characteristics found in historical recordings. Therefore, we alter each noise segment replica before the OLA synthesis step in two ways: i) applying a random perturbation to the phase of the noise segment (adding Gaussian noise $\sim \mathcal{N}(0, 0.1)$ to the phase of the STFT); ii) applying a random shift in time (with wraparound). We found that these simple operations produce longer noise samples with auditory characteristics similar to the ones encountered in the historical recordings, avoiding artificial periodic patterns.

Finally, we create time-aligned pairs of <clean, noise> examples by: i) applying band-pass filtering with cut-off frequencies randomly sampled in [50Hz, 150Hz] and [5kHz, 10kHz], respectively; ii) mixing a randomly selected noise sample with a gain in the range [10dB, 30dB].

2.2 Model architecture

The generator processes the audio in the time-frequency domain. It first computes the STFT of the input, the real and imaginary components of which are then fed as a 2-channel image to a 2D convolutional U-Net [23] followed by an inverse STFT back to the time domain. Finally the output is added back to the input, making the model a residual generator.

The U-Net in our generator is a symmetric encoder-decoder network with skip-connections, where the architecture of the decoder layers mirrors that of the encoder and the skip-connections run between each encoder block and its mirrored decoder block. Each encoder block is a 3×3 convolution followed by either a 3×4 convolution with stride of 1×2 (if down-sampling in the frequency dimension), or a 4×4 convolution with stride of 2×2 (if down-sampling in both time and frequency dimensions). We choose kernel sizes to be multiples of strides to ensure even contribution from all locations of the input feature map, which prevents the formation of checkerboard-like patterns in resampling layers [24]. The decoder blocks mirror the encoder blocks, and each consists of a transposed convolution for up-sampling followed by a 3×3 convolution. Each decoder block additionally includes a shortcut connection between its input and output. The shortcut consists of a nearest-neighbor up-sampling layer, which

[1] https://www.youtube.com/playlist?list=PLa5CkN3odpnxi3WqMH4MgVk7XUjCP99d3

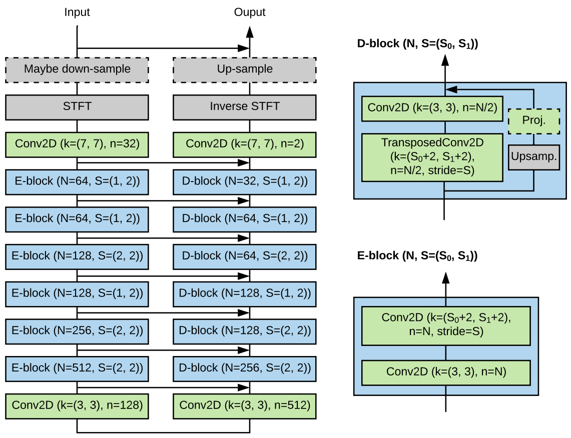

Figure 1. Generator architecture. Dashed-line components are included on a need-to-have basis: Up/down-sampling of the input/output audio is needed for processing at coarser resolutions in a multi-scale setup; The linear projection (by 1x1 convolution) in the decoder block is present only when the output of the block has a different number of channels from its input.

is followed by a linear projection using 1x1 convolution when the output has a different number of channels from the input. We do not include a shortcut in the encoder block, since it already shares the same input with a U-Net skip connection and therefore only needs to produce the residual complementary to the skip path. The architecture of the generator is shown in Figure 1.

We use two discriminators for the adversarial objective, one in the waveform domain and one in the STFT domain. The STFT discriminator has the same architecture as the encoder module of the generator. For the waveform discriminator, we use the same architecture as MelGAN [25] except that we only double (instead of quadruple) the number of channels in the down-sampling layers. We found this light-weight version to be sufficient in our setup, and that using the full version had no additional benefit. Both discriminators are fully convolutional. Hence the waveform discriminator produces a 1D output spanning the time domain, and the STFT discriminator has a 2D output spanning the time-frequency domain.

We use weight normalization [26] and ELU activation [27] in the generator, while layer normalization [28] and Leaky ReLU activation [29] with $\alpha = 0.3$ are used in the discriminator.

2.2.1 *STFT Representation*

In the generator, the STFT is represented by a 2-channel image, where the channels are the real and imaginary components. We also explored a polar representation, where the channels are the modulus and the phase; additionally we experimented with processing only the modulus channel and reusing the original phase, as is done in [30]. Nevertheless, we found the real/imaginary representation to perform better in our experiments.

Furthermore, we tried aligning the phase so that the phase in each frame is coherent with a global reference (e.g., the first frame) rather than its local STFT window. Again, we observed no advantage in doing so, which suggests that the neural network is capable of internally handling the phase offsets. Unlike [30], we do not convert STFT to logarithmic scale as we found it be detrimental to performance (even with various smoothing and normalization schemes).

2.2.2 *Multi-scale Generator*

We can further stack multiple copies of the generator described above, each with its own separate parameters, in a coarse-to-fine fashion: The generators at earlier stages process the audio at reduced temporal resolutions, whereas the later-stage generators focus on restoring finer details. This is equivalent to halving the sampling rate in each scale. This type of multi-scale generation scheme is routinely used in computer vision and graphics to produce high-resolution images (e.g., [31]).

Let K be the total number of scales, then generator G_k at scale k ($k \in \{0, \dots, K-1\}$) down-samples its input by a factor of 2^k before computing the STFT and up-samples the output residual (after computing the inverse STFT) by the same factor to match the resolution of the input. The overall generator G is the composite of $G_0 \circ \cdots \circ G_{K-1}$.

Compared with simply stacking U-Nets all at the original input resolution, as done in [32], the benefit of the multi-scale approach is two-fold: i) the asymptotic computational complexity is constant with respect to the number of scales, as opposed to linear in [32], due to exponentially decreasing input sizes at coarser levels; ii) the intermediate outputs of the generator correspond to the input audio processed at lower resolutions, which allows us to meaningfully impose multi-scale losses on the intermediate outputs in addition to the final output. We will describe how this can be accomplished in the next section.

2.3 Training

The generator can be trained using the reconstruction loss between the denoised output and the clean target. This can be further complemented with an adversarial loss, given by discriminators trained simultaneously with the generator, a practice often used in audio enhancement (e.g., [18,20,30], among others). In the case of our multi-scale generator, we use the same number of waveform and STFT discriminators as generator scales. This way, there is one discriminator of both types for each of the (down-sampled) intermediate outputs and final output in each domain. For the adversarial loss, we use the hinge loss averaged over multiple scales. Since the discriminators are convolutional, this loss is further averaged over time for the waveform discriminator and over time-frequency bins for the STFT discriminator. Similarly, the reconstruction loss is also imposed on the outputs at each scale.

More formally, let (x, y) denote a training example, where x is the noisy input and y is the clean target, and $k \in \{0, \dots, K-1\}$ denote the scale index. Hence y_k is the clean audio down-sampled to scale k, and $\hat{y}_k$ represents the intermediate output of the generator $G_k \circ \cdots \circ G_{K-1}(x)$ down-sampled to the same scale. Note that for the finest scale $k = 0$ at full resolution, $y_0 = y$ is simply the original clean audio and $\hat{y} \triangleq \hat{y}_0 = G(x)$ is the final output of the generator. Thus the L^1 reconstruction loss in the STFT domain can be written as

$$\mathcal{L}_G^{\text{rec}} = \mathbb{E}_{(x,y)}\left[\sum_k \frac{\|\omega_k - \hat{\omega}_k\|_1}{S_k^{\text{STFT}}}\right], \tag{1}$$

where 2D complex tensors ω_k and $\hat{\omega}_k$ denote the STFT of down-sampled clean audio y_k and generator output $\hat{y}_k$ for scale k, respectively, and S_k^{STFT} is the total number of time-frequency bins in ω_k and $\hat{\omega}_k$. We find this STFT-based reconstruction loss to perform better than either imposing per-sample losses directly in the waveform domain or using losses computed from the internal "feature" layers of discriminators (e.g. [25]).

For the adversarial loss, let t denote the temporal index over all T_k logits of the waveform discriminator at scale k (recalling that the discriminators are fully convolutional) and let s denote the index over all S_k logits of the STFT discriminator. Then discriminator losses in the wave and STFT domains can be written as, respectively,

$$\begin{aligned}\mathcal{L}_D^{\text{wave}} = \; & \mathbb{E}_y\left[\sum_{k,t} \frac{1}{T_k}\max(0, 1 - D_{k,t}^{\text{wave}}(y_k))\right] + \\ & \mathbb{E}_x\left[\sum_{k,t} \frac{1}{T_k}\max(0, 1 + D_{k,t}^{\text{wave}}(\hat{y}_k))\right] \end{aligned} \tag{2}$$

$$\begin{aligned}\mathcal{L}_D^{\text{STFT}} = \; & \mathbb{E}_y\left[\sum_{k,s} \frac{1}{S_k}\max(0, 1 - D_{k,s}^{\text{STFT}}(y_k))\right] + \\ & \mathbb{E}_x\left[\sum_{k,s} \frac{1}{S_k}\max(0, 1 + D_{k,s}^{\text{STFT}}(\hat{y}_k))\right], \end{aligned} \tag{3}$$

and the corresponding adversarial loss for the generator is given by

$$\begin{aligned}\mathcal{L}_G^{\text{adv}} &= \mathcal{L}_G^{\text{adv, wave}} + \mathcal{L}_G^{\text{adv, STFT}} \\ &= \mathbb{E}_x\left[\sum_{k,t} \frac{1}{T_k}\max(0, 1 - D_{k,t}^{\text{wave}}(\hat{y}_k)) + \right. \\ & \qquad \left. \sum_{k,s} \frac{1}{S_k}\max(0, 1 - D_{k,s}^{\text{STFT}}(\hat{y}_k))\right]. \end{aligned} \tag{4}$$

The overall generator loss is a weighted sum of the adversarial loss and the reconstruction loss, i.e.,

$$\mathcal{L}_G = \mathcal{L}_G^{\text{rec}} + \lambda \cdot \mathcal{L}_G^{\text{adv}}. \tag{5}$$

We set the weight of the adversarial loss λ to 0.01 in all our experiments, except those where we do not use discriminators (which corresponds λ=0). We train the model with TensorFlow for 400,000 steps using the ADAM [33] optimizer, with a batch size of 16 and a constant learning rate of 0.0001 with $\beta_1 = 0.5$ and $\beta_2 = 0.9$. For the STFT, we use a window size of 2048 and a hop size of 512 when there is only a single scale. For each added scale we halve the STFT window size and hop size *everywhere*. This way the STFT window at the coarsest scale has a receptive field of 2048 samples at the original resolution, whereas finer levels have smaller receptive fields and hence focus more on higher frequencies.

Our model has around 9 million parameters per scale in the generator. At inference-time, it takes less than half a second for every second of input audio on a modern CPU and more than an order of magnitude faster on GPUs.

3. EXPERIMENTS

We evaluate our model on a dataset of synthetically generated noisy-clean pairs, using both objective and subjective metrics. In addition, we also provide a subjective evaluation on samples from real historical recordings, for which the clean references are not available.

3.1 Datasets

Our data is derived from two sources: i) digitized historical music recordings from the Public Domain Project [34], and ii) a collection of classical music recordings of CD-quality. The historical recordings are used in two ways: i) to extract realistic noise from relatively silent portions of the audio, as described in Section 2.1; and ii) to evaluate different methods based on the human-perceived subjective quality of their outputs. The modern recordings are used for mixing with the extracted noise samples to create synthetic noisy music, as well as serving as the clean ground truth. We additionally filter our data to retain only classical music, as it is by far the most represented genre in historical recordings. The resulting dataset consists of pairs of clean and noisy audio clips, both monophonic and 5 seconds long, sampled at 44.1kHz. The total duration of the clean clips is 460h.

3.2 Quantitative Evaluation

We quantitatively evaluate the performance of different methods on a held-out test set of 1296 examples from the synthetic noisy music dataset. For the neural network models, whose training is stochastic, we repeat the training process 10 times for each model and report the mean for each metric and its standard error.

Evaluation metrics: Objective metrics such as the signal-to-noise ratio (SNR) faithfully measure the difference between two waveforms on a per-sample basis, but they often do not correlate well with human-perceived reconstruction quality. Therefore, we additionally measure the *VGG distance* between the ground truth and the denoised output, which is defined as the L^2 distance between their respective embeddings computed by a VGGish network [35]. The embedding network is pre-trained for multi-label classification tasks on the YouTube-100M dataset, in which labels are assigned automatically based

	ΔSNR (dB)	-ΔVGG
1 scale	**3.4±0.0**	0.68±0.01
2 scales	**3.4±0.0**	**0.78±0.01**
3 scales	3.2±0.0	0.73±0.01

Table 1. Performance of our model with different numbers of scales K in terms of SNR gain (ΔSNR) and VGG distance reduction (-ΔVGG). Higher is better.

on a combination of metadata (title, description, comments, etc.), context, and image content for each video. Hence we expect the VGG distance to focus more on higher-level features of the audio and less on per-sample alignment. Note that the same embedding used by Frechét audio distance (FAD) [36], which measures the distance between two *distributions*. However, FAD does not compare the content of individual audio samples, and is hence not applicable to denoising.

We report the SNR gain (ΔSNR) and VGG distance reduction (-ΔVGG) of the denoised output relative to the noisy input, averaged over the test set. For reference, the noisy input has an average SNR of 14.4dB and VGG distance of 2.09. Table 1 shows the performance of our model with different numbers of scales. We use $K = 2$ scales for the rest of our experiments. We evaluate variants of our proposed model in an ablation study and compare with alternative approaches and well-established signal processing baselines:

- **Ours, λ=0**: Our model trained with only reconstruction loss.
- **Ours, λ=0.01**: Our model trained with both adversarial and reconstruction losses.
- **Ours, bypass phase**: Same as above, except that the phase of the noisy input is reused and only the modulus of the STFT is processed by the U-Net (as a single-channel image). This is similar to the approach of [30], but trained and evaluated for music denoising instead of speech.
- **MelGAN-UNet**: A 1D-convolutional waveform-domain generator inspired by MelGAN [25], where the decoder is the same as the generator of MelGAN and the encoder mirrors the decoder.
- **DeepFeature generator**: The 1D-convolutional waveform-domain generator of [17], which does not use U-Net but rather a series of 1D convolutions with exponentially increasing dilation sizes. Unlike U-Net, the temporal resolution and number of channels remain unchanged in all layers of this network.
- **log-MMSE**: A short-time spectral amplitude estimator for speech signals which minimizes the mean-square error of the log-spectra [37]. In our implementation, the estimation of the noise spectrum is based on low-energy frames across the whole clip, rather than considering the frames at the start of the audio clip. We use this deviation from the standard implementation as it gives better SNR results.
- **Wiener**: A linear time-invariant filter that minimizes the mean-square error. We adopted the SciPy [38] implementation and used default parameters, as different parameter settings did not improve the results.

For waveform-domain generators, we tried waveform-domain losses – including reconstruction losses in the "feature space" of discriminator internal layers [17, 25] – as well as STFT-domain losses, and found the former to work better with the DeepFeature generator while the latter gave better results for the MelGAN-UNet generator. The results shown for these generators are those obtained with the better loss variant. We also divide the test set into three subsets, each containing the same number of examples, with low noise (avg. 19.8dB SNR), medium noise (avg. 14.2dB SNR), and high noise (avg. 9.4dB SNR), and compute the same metrics on each subset as well as on the full test set.

The results in Table 2 show that, for all noise levels, our model consistently outperforms the signal processing baselines and the waveform-domain neural network models, which have proven highly successful in speech enhancement but are not adequate for the complexity of music signals. The signal-processing baselines (log-MMSE and Wiener filtering) are hardly able to improve upon the noisy input at all. This is not too surprising given the non-Gaussian, non-white nature of the real-world noise in the evaluation data. Comparing the results among the variants of our model, we further make the following observations:

- Using adversarial losses does not help in terms of SNR, as is evident from the top two rows of Table 2. The SNR decrease is small but significant. The adversarially trained variant, however, scores better on the high-level feature oriented VGG distance metric, which is in line with past observations [18, 25]
- It is advantageous to take both the modulus and the phase into account when processing the STFT spectrogram, as the "bypass-phase" variant which reuses the input phase produces consistently worse results across all noise levels. This shows that the proposed model is able to reconstruct the fine-grained phase component of the original clean music.

3.3 Subjective Evaluation

In the previous section we compared results by means of objective quality metrics, which can be quantitatively computed from pairs of noisy-clean examples. These metrics can be conveniently used to systematically run an evaluation over a large number of samples. However, it is difficult to come up with an objective metric that correlates with quality as perceived by human listeners. Indeed, the SNR and VGG distance metrics do not agree in our quantitative evaluation – the proposed model is better in terms of VGG distance, but worse in terms of SNR compared to its counterpart without discriminator. We now describe our

	ΔSNR (dB)				-ΔVGG			
	noise level				noise level			
	low	medium	high	all	low	medium	high	all
Ours, λ=0	**2.5**±**0.0**	**4.1**±**0.0**	**4.3**±**0.0**	**3.7**±**0.0**	0.30±0.01	0.47±0.01	0.58±0.01	0.45±0.01
Ours, λ=0.01	2.2±0.0	3.9±0.0	4.1±0.0	3.4±0.0	**0.66**±**0.01**	**0.81**±**0.01**	**0.87**±**0.01**	**0.78**±**0.01**
Ours, bypass phase	2.1±0.0	3.5±0.0	3.7±0.0	3.1±0.0	0.62±0.01	0.77±0.01	0.83±0.01	0.74±0.01
MelGAN-UNet	1.7±0.0	2.9±0.0	3.1±0.0	2.6±0.0	0.16±0.02	0.15±0.03	0.18±0.02	0.16±0.02
DeepFeature generator	-0.7±0.4	1.3±0.1	1.7±0.1	0.8±0.2	0.00±0.02	0.03±0.02	0.00±0.01	0.01±0.02
log-MMSE	-1.4	-0.2	0.1	-0.5	-0.15	-0.04	0.01	-0.07
Wiener	0.1	0.1	0.1	0.1	0.01	0.02	0.01	0.01

Table 2. Performance of different variants of our model and alternative approaches, evaluated on subsets of examples with different noise levels as well as on the full test set.

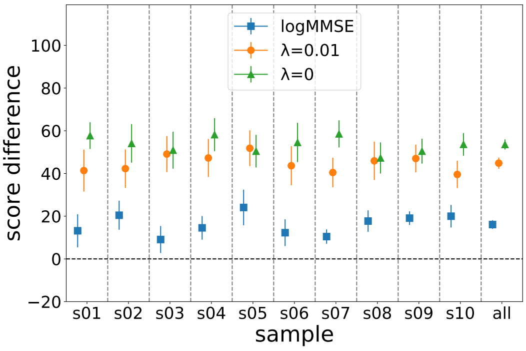

Figure 2. Average score differences for the historical recordings dataset, relative to the original noisy sample.

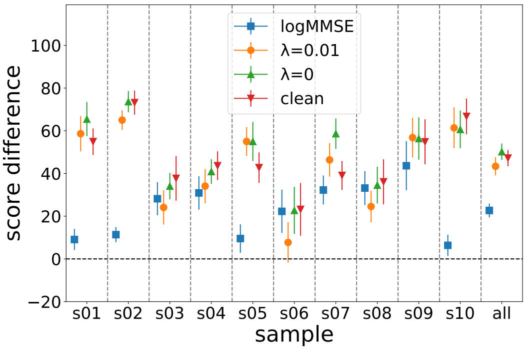

Figure 3. Average score differences for the synthetic dataset, relative to the noisy sample.

subjective evaluation which we ran in order to identify the method that performs best when judged by humans.

Following recent work on low-bitrate audio improvement [21], we use a score inspired by MUSHRA [22] for our subjective evaluation. Each rater assigned a score between 0 and 100 to each sample. The main difference to actual MUSHRA scores is that since no clean reference exists for historical recordings, we do not include an explicit reference in the rated samples (although we do include the clean sample in the synthetic dataset evaluation).

We perform our evaluation on 10 samples of historical recordings, and separately on 10 samples from the synthetic dataset, using 11 human raters. As in the objective evaluation, each sample is 5 seconds long. We evaluate the following four versions for each sample: (i) Original historic audio example, (ii) denoised example using our model with λ=0.01, (iii) denoised example using our model with λ=0, (iv) denoised example using log-MMSE.

For the synthetic dataset, we use the four versions above, but instead of the historic audio we use the synthetically noisified one. We do not include Wiener filtering as a competing baseline here since we noticed that it produces outputs that are consistently near-identical to the noisy input, and hence including it in the subjective evaluation would provide little value. We use the original noisy audio as the reference from which to compute score differences for the historical recordings, and the synthetically noisified sample as the reference for the synthetic data. The results are shown in Figure 2 for the historical recordings, and in Figure 3 for the synthetic dataset. Error bars are 95% confidence intervals, assuming a Gaussian distribution of the mean. Both of our methods significantly improve the historical recordings, by around 50 points on average. In comparison, the logMMSE baseline only improves by an average of 16 points. We also performed a Wilcoxon signed-rank test between our λ=0.01 and λ=0 models, to find that the difference is statistically significant (p-value $< 1.19 \times 10^{-11}$). On the synthetic data, again the λ=0 model outperforms the λ=0.01 variant, with a p-value $< 2.13 \times 10^{-8}$. On the other hand, there is no significant difference between the mean score differences of the λ=0 model and the clean sample (p-value = 0.097).

4. CONCLUSION

We presented a learning-based method for automated denoising and applied it to restoration of noisy historical music recordings, matching a high quality bar: Judged by human listeners on actual historical records, our method improves audio quality by a large margin and strongly outperforms existing approaches on a MUSHRA-like quality metric. On artificially noisified music, it even attains a quality level that listeners found to be statistically indistinguishable from the ground truth.

5. REFERENCES

[1] S. H. Godsill and P. J. Rayner, *Digital Audio Restoration: A Statistical Model Based Approach*, 1st ed. Berlin, Heidelberg: Springer-Verlag, 1998.

[2] M. Berouti, R. Schwartz, and J. Makhoul, "Enhancement of speech corrupted by acoustic noise," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 4, 1979, pp. 208–211.

[3] S. Kamath and P. Loizou, "A multi-band spectral subtraction method for enhancing speech corrupted by colored noise," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 4, 05 2002.

[4] A. M. Reddy and B. Raj, "Soft mask methods for single-channel speaker separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 6, pp. 1766–1776, 2007.

[5] E. M. Grais and H. Erdogan, "Single channel speech music separation using nonnegative matrix factorization and spectral masks," in *International Conference on Digital Signal Processing (DSP)*, 2011, pp. 1–6.

[6] P. Scalart and J. V. Filho, "Speech enhancement based on a priori signal to noise estimation," in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol. 2, 1996, pp. 629–632.

[7] H. Attias, J. C. Platt, A. Acero, and L. Deng, "Speech denoising and dereverberation using probabilistic models," in *Advances in Neural Information Processing Systems 13*, 2001, pp. 758–764.

[8] P. C. Loizou, "Speech enhancement based on perceptually motivated bayesian estimators of the magnitude spectrum," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, pp. 857–869, 2005.

[9] C. Fevotte, B. Torresani, L. Daudet, and S. J. Godsill, "Sparse linear regression with structured priors and application to denoising of musical audio," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 174–185, 2008.

[10] J. Deng, B. W. Schuller, F. Eyben, D. Schuller, Z. Zhang, H. Francois, and E. Oh, "Exploiting time-frequency patterns with LSTM-RNNs for low-bitrate audio restoration," *Neural Computing and Applications*, vol. 32, no. 4, pp. 1095–1107, 2020. [Online]. Available: https://doi.org/10.1007/s00521-019-04158-0

[11] N. Bassiou, C. Kotropoulos, and I. Pitas, "Greek folk music denoising under a symmetric α-stable noise assumption," in *10th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, 2014, pp. 18–23.

[12] V. Mach, "Denoising phonogram cylinders recordings using structured sparsity," in *2015 7th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, 2015, pp. 314–319.

[13] N. Perraudin, N. Holighaus, P. Majdak, and P. Balazs, "Inpainting of long audio segments with similarity graphs," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. PP, pp. 1–1, 02 2018.

[14] V. Kuleshov, S. Z. Enam, and S. Ermon, "Audio super resolution using neural networks," in *5th International Conference on Learning Representations (ICLR) 2017, Workshop Track, Toulon, France*, 2017.

[15] S. Birnbaum, V. Kuleshov, Z. Enam, P. Koh, and S. Ermon, "Temporal film: Capturing long-range sequence dependencies with feature-wise modulations," in *Proc. 33rd Annual Conference on Neural Information Processing Systems (NeurIPS 2019)*, 2019.

[16] M. Michelashvili and L. Wolf, "Audio denoising with deep network priors," *CoRR*, vol. abs/1904.07612, 2019. [Online]. Available: http://arxiv.org/abs/1904.07612

[17] F. G. Germain, Q. Chen, and V. Koltun, "Speech denoising with deep feature losses," 2018.

[18] S. Pascual, A. Bonafonte, and J. Serrà, "SEGAN: Speech enhancement generative adversarial network," in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association*, 08 2017, pp. 3642–3646.

[19] D. Rethage, J. Pons, and X. Serra, "A wavenet for speech denoising," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5069–5073.

[20] C. Donahue, B. Li, and R. Prabhavalkar, "Exploring speech enhancement with generative adversarial networks for robust speech recognition," *CoRR*, vol. abs/1711.05747, 2017. [Online]. Available: http://arxiv.org/abs/1711.05747

[21] A. Biswas and D. Jia, "Audio codec enhancement with generative adversarial networks," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2020.

[22] "Method for the subjective assessment of intermediate quality levels of coding systems ITU-Recommendation BS.1534-3," 2015. [Online]. Available: www.itu.int/rec/R-REC-BS.1534

[23] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.

[24] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, 2016. [Online]. Available: http://distill.pub/2016/deconv-checkerboard

[25] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brebisson, Y. Bengio, and A. Courville, "MelGAN: Generative adversarial networks for conditional waveform synthesis," 2019.

[26] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 901–909.

[27] S. H. Djork-Arné Clevert, Thomas Unterthiner, "Fast and accurate deep network learning by exponential linear units (elus)," in *ICLR*, 2016.

[28] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[29] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

[30] S. Abdulatif, K. Armanious, K. Guirguis, J. T. Sajeev, and B. Yang, "Aegan: Time-frequency speech denoising via generative adversarial networks," 2019.

[31] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. [Online]. Available: https://openreview.net/forum?id=Hk99zCeAb

[32] K. Armanious, C. Yang, M. Fischer, T. Küstner, K. Nikolaou, S. Gatidis, and B. Yang, "Medgan: Medical image translation using GANs," *CoRR*, vol. abs/1806.06397, 2018. [Online]. Available: http://arxiv.org/abs/1806.06397

[33] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.

[34] "Public domain project," http://pool.publicdomainproject.org, [Online; accessed February-2020]. [Online]. Available: http://pool.publicdomainproject.org

[35] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. Weiss, and K. Wilson, "CNN architectures for large-scale audio classification," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017. [Online]. Available: https://arxiv.org/abs/1609.09430

[36] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, "Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms," in *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, G. Kubin and Z. Kacic, Eds. ISCA, 2019, pp. 2350–2354. [Online]. Available: https://doi.org/10.21437/Interspeech.2019-2219

[37] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, no. 2, pp. 443–445, 1985.

[38] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. Jarrod Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. Carey, İ. Polat, Y. Feng, E. W. Moore, J. Vand erPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and S. . . Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.