Article

Characterization of Deep-Learning-Based Speech Enhancement Techniques in Online Audio Processing Applications

Caleb Rascon [1]

[1] Instituto de Investigaciones en Matematicas Aplicadas y en Sistemas, Universidad Nacional Autonoma de Mexico; caleb@unam.mx

Abstract: Deep-learning-based speech enhancement techniques have been recently grown in interest, since their impressive performance can potentially benefit a wide variety of digital voice communication systems. However, such performance has been evaluated mostly in offline audio processing scenarios (i.e. feeding the model, in one go, a complete audio recording, which may extend several seconds). It is of great interest to evaluate and characterize the current state-of-the-art in applications that process audio online (i.e. feeding the model a sequence of segments of audio data, concatenating the results at the output end). Although evaluations and comparisons between speech enhancement techniques have been carried out before, as far as the author knows, the work presented here is the first that evaluates the performance of such techniques in relation to their online applicability. Meaning, this work measures how the output signal-to-interference ratio (as a separation metric), the response time and memory usage (as online metrics) are impacted by the input length (the size of audio segments), in addition to the amount of noise, amount and number of interferences, and amount of reverberation. Three popular models were evaluated, given their availability on public repositories and online viability: MetricGAN+, Spectral Feature Mapping with Mimic Loss, and Demucs-Denoiser. The characterization was carried out using a systematic evaluation protocol based on the Speechbrain framework. Several intuitions are presented and discussed, and some recommendations for future work are proposed.

Keywords: speech enhancement; online applicability; real-time factor

1. Introduction

Speech is captured in real-life scenario along with noise and 'interferences' (other sources that are not of interest). The captured audio signal can be considered as an additive mixture of several audio 'streams', one of which is the original stream of the target speech. To this effect, speech enhancement is the removal of such noise and interferences from a given audio signal, such that only the target speech remains in the resulting enhanced audio signal. It has recently grown in popularity thanks in large part to the recent surge in performance of deep-learning-based models [1]. Speech enhancement is closely related to 'speech separation' [2], since they bare similar objectives. However, in this work a distinction is made between the two, since the latter aims to separate all of the speech streams in the audio signal into their own audio separate 'channel', while the former aims to "separate" only the target speech. Still, both sets of techniques share several frames of reference, specifically, how to model speech and separate it from other audio streams in the mixture.

The types of speech enhancement techniques can be divided in several ways, but the one relevant to this work is in the way the captured input audio signal is being fed to the technique. 'Offline' speech enhancement techniques are fed the entire audio recording and, correspondingly, estimate all the target speech in one go. This type of techniques are of interest in applications such as security by voice [3], audio refinement for musical recordings [4], and movie soundtrack enhancement [5]. 'Online' speech enhancement techniques are sequentially fed audio segments of the captured input audio signal through time (usually coming directly from a live microphone) and, correspondingly, aim to estimate the target speech in each audio segment, which can then be outputted directly to a speaker,

Citation: Rascon, C. Characterization of Deep-Learning-Based Speech Enhancement Techniques in Online Audio Processing Applications. *Preprints* **2023**, *1*, 0. https://doi.org/

with some expected latency[1]. This type of techniques are of interest in applications such as real-time automatic speech recognition [6], sound source localization in service robotics [7], hearing prosthesis [8], mobile telecommunication [9], and video-conferencing [10].

Although there have prior important and thorough comparisons between speech enhancement techniques [11–14], they were carried out evaluating the techniques in an offline manner. As far as the author knows, there hasn't been a comparison between speech enhancement techniques with a focus on online applicability.

To this effect, the objective of this work is to characterize a representative sample of online speech enhancement techniques, chosen because of their ease of replicability, online viability and popularity. The characterization described in this work is based on a systematic evaluation, using a synthesized evaluation corpus by 'disturbing' a sub-set of recordings from the LibriSpeech corpus [15]. These disturbances were: 1) adding noise at varying amounts of signal-to-noise ratio; 2) adding varying amounts of interferences with varying amounts of signal-to-interference ratio; and, 3) adding reverberation at varying amounts of scaling factors. The 'online' aspect of this characterization was carried by feeding the techniques a sequence of audio segments extracted from the synthesized recordings, and then concatenating the results. The lengths of the audio segments (or 'input lengths') were varied throughout the evaluation process, but kept the same for each evaluation iteration. The enhancement performance was measured by means of the output signal-to-noise ratio, and registered for each evaluation iteration, along with the inference response time and memory usage through time.

The techniques that were characterized in this manner were Spectral Feature Mapping with Mimic Loss [16], MetricGAN+ [17], and Demucs-Denoiser [18]. The main motivation behind this characterization is to provide insights to the reader about what to expect from the current state-of-the-art of online speech enhancement.

This works is organized as follows: Section 2 presents a brief summary of the three chosen techniques to be characterized; Section 3 details the evaluation methodology, specifically, how the evaluation corpus was created and how the evaluation was carried out systematically; Section 4 presents the results; Section 5 provides some insights and recommendations, based on the previously presented results; and, Section 6 provides some concluding remarks and proposals for future work.

2. Evaluated Techniques

As mentioned beforehand, the objective of this work is to characterize the online applicability of current speech enhancement state of the art. Recently, there have been major advances in terms of separation performance [1] and the perceptual quality of the enhanced speech [19,20]. However these advances have been mostly carried out in offline processing scenarios, while, until very recently [16–18], relatively little attention has been given to online speech enhancement. It is not the objective of this work to argue why this is the case[2], and it would be unfair to evaluate techniques in scenarios which they weren't designed for. However, this limits the amount of models that can be evaluated, even more so considering that few of them provide ways to easily replicate their results. Thus, the selection was based on the following criteria:

1. *Ease of replicability*: the existence of a public code repository to recreate its published results.
2. *Online viability*: it can be deduced from the original published paper that it is able to run in an online manner.
3. *Popularity*: if a pre-trained model is publicly available in one or more speech enhancement repositories (such as Hugging Face [21]) to be downloaded and tested.

[1] Latency is the time difference between the moment the audio segment is fed to the technique and the estimated target speech in that audio segment is reproduced to a speaker.

[2] Although, to be fair, it is a very challenging scenario, resulting from the absence of data to "look ahead" to, as well as the harsh time constraints. Both of these, in conjunction, may have contributed to this oversight.

 doi:10.20944/preprints202303.0158.v1

To this effect, three models were selected to be evaluated:

- Spectral Feature Mapping with Mimic Loss [16]
- MetricGAN+ [17]
- Demucs-Denoiser [18]

It is important to mention that this list is not to be considered exhaustive, but as a representative sample of the current state of online speech enhancement. In this section, the three techniques are described, for completeness sake.

2.1. Spectral Feature Mapping with Mimic Loss

The intent of the original paper [16] was not to create a speech enhancement model, but to introduce a loss function that could be used to train speech enhancement models to make Automatic Speech Recognizers (ASRs) more robust. Until that point, it was common to refine ASRs models with the output of the speech enhancement technique since, although they were very good at removing noise and interferences, they also distorted the signal (albeit lightly) and/or inserted artifacts, both of which degraded the ASR performance. To this effect, a loss function was designed to create a training objective that estimated the target speech with high perceptual quality, not just well separated from the rest of the audio streams in the captured audio signal.

To do this, first, a spectral classifier was trained using a cross-entropy loss function (referred to as 'classification loss', L_c) that aimed to classify from clean speech the presence of similar acoustic events (or 'senones'). In parallel, a spectral mapper was trained using a mean-square error loss function (referred to as 'fidelity loss', L_f) that aimed to map clean speech from noisy speech. Then, with the weights of the spectral classifier frozen, the spectral mapper was further trained but now using the fidelity loss function in conjunction with another loss function (referred to as 'mimic loss', L_m): the mean square difference between the senones classified directly from a reference clean speech signal, and those that were classified from the estimated target speech. In this regard, the fidelity loss function represents the objective of de-noising speech, while the mimic loss represents the objective of providing high perceptual quality of the de-noised speech. In conjunction, solving for both losses at the same time (as a simple weighted sum), should result in highly de-noised, highly perceptual speech, appropriate to be used directly on pre-trained ASRs.

To provide evidence of the usability of the proposed loss function, the authors trained a simple spectral mapper that consisted of a two-layer feed-forward network, with only 2048 neurons in each layer. Because of the simplicity of their proposed model, low memory requirements and small response time can be assumed.

2.2. MetricGAN+

From the explanation of the last technique, it could be deduced (which the authors themselves admit to) that there is a sort of inspiration of Generative Adversarial Networks (GAN) in the design of the mimic loss. This is because there is a type of "adversarial" scheme between the two trained models, from which a result is generated. This is basically the idea behind GANs [22]: a 'generator network' that is trained and "put against" the result of an 'adversarial network', to generate a result. To this effect, MetricGAN [23] was introduced as a way to directly integrate into a loss function some metrics for evaluating perceptual quality, such as the perceptual evaluation of speech quality (PESQ) [24] and short-time objective intelligibility (STOI) [25]. This was proposed so that a network that uses such a loss function, would result in it mimicking the aforementioned metrics. Consequently, this was used to train the 'discriminator network' of the GAN; the 'generator network' of the GAN was trained solely relying on the adversarial loss function, as it would normally.

Later, MetricGAN+ [23] was introduced as a refinement of MetricGAN, with the following improvements:

- In addition to learning the metric behavior from the clean speech and estimated speech, it was also learned from the noisy speech, stabilizing the learning process of the discriminator network (which aims to mimic such metrics).

 doi:10.20944/preprints202303.0158.v1

- Generated data from previous epochs was re-used to train the discriminator network, improving performance.
- Instead of using the same sigmoid function for all frequencies in the generator network, a sigmoid function was learned for each frequency, providing a closer match between the sum of the clean and noise magnitudes, and the magnitude of the noisy speech.

The authors trained a two-layered bi-directional long-short-term memory (BLSTM) network, with only 200 neurons in each layer, followed by two fully connected layers, a ReLU layer and the learned sigmoid layer. The size of the network is bigger than the one trained form SFM-Mimic, but it is still small, from which the same assumption of memory requirements and response times can be made.

2.3. Demucs-Denoiser

A sizable part of the speech enhancement techniques that are currently in literature aim to solve the problem by estimating a frequency mask that, when applied to the input signal, results in an enhanced output [2]. This implies that there is a considerable part of the training process that needs to be focused in classifying each time-frequency bin as either part of the target speech or not. This may not only require a high amount of memory resources, but the transformation to and from the frequency domain may add on to the response time of the model. To this effect, the Demucs model [26] was introduced first to separate music instruments from musical recordings, directly in the time domain. The Demucs-Denoiser model [18] is an evolution of such model focused on speech enhancement, which from a musical point of view, aims to separate one "instrument" (target speech) from the rest of the "recording" (noisy speech).

The Demucs model network employs a type of encoder-decoder architecture, with skip connections between symmetric parts of the encoding and decoding parts of the model network. This structure is also known as a U-Net structure, which was first used for biomedical imaging [27]. Once the audio data is encoded, causal prediction is carried out to de-noise it at an encoded level. Then, this de-noised encoded data is decoded (with the help of the skip connections) to produce the estimated target speech waveform. The authors also employed an up-sampling scheme, with which the encoding-decoding process provided good results. Additionally, for training purposes, the authors employed two loss functions between the clean and estimated target speech: an L1 loss function over the waveform domain, and an STFT loss over the time-frequency domain.

The encoding part of the model was conformed by 5 meta-layers. Each, in turn, was conformed by a 1-dimensional convolutional layer with a ReLU layer, and another convolutional layer with a GRU layer. The decoder part was made up of the same meta-layers but in reverse, each with its respective skip connection. The causal prediction was carried out by a 2-layer LSTM network. The resulting model may appear to require a relatively large amount of memory (as opposed to the other two techniques) which, in turn, may imply large response times. However, the authors verified that it was able to run in an online manner, although the input lengths were not as varied as it is presented in this work.

The authors also trained the model with different training data sets, producing different versions of their models. The model characterized in this work is the one trained with the DNS dataset [28], with 64 initial output convolutional channels.

3. Evaluation Methodology

The objective of this work is to characterize speech enhancement techniques under different conditions that are applicable to online processing scenarios. Thus, in addition to evaluating techniques in the presence of different levels of noise, different levels of reverberation and different amounts of speech interferences, the evaluation is to be carried out using different input lengths. To this effect, an evaluation corpus was synthesized for this purpose, and a windowing methodology was implemented that simulates what these techniques should be put through when carrying out speech enhancement in an online manner. Finally, when evaluating the techniques, several metrics are measured that are

 doi:10.20944/preprints202303.0158.v1

relevant to the quality of the result (mainly signal-to-interference ratio) as well as to the implementation viability (such as response time and memory consumption.)

3.1. Input Lengths and Online Processing

It is crucial to subject the speech enhancement techniques to different input lengths (N_w) in a systematic and repeatable manner, so that the comparisons between techniques are "apples-to-apples". Thus, given an specific input length, a given recording (the manner in which it is synthesized is detailed in Section 3.2) is divided into subsequent audio segments of that length. The last segment is zero-padded if its length is smaller than the given input length.

Each audio segment is then fed to the speech enhancement technique, chronologically. The audio segments that the technique outputs are concatenated, in the sequence they were fed. Finally, the result of this concatenation is sliced so that it is the same length of the given recording, removing the zeroes that were padded to the last segment, providing the final estimated output. From this output, the performance metrics are calculated (detailed in Section 3.4.1).

3.2. Corpus Synthesis, Evaluated Variables

As mentioned beforehand, a corpus is required to carry out the characterization of the speech enhancement techniques. To this effect, the 'dev-clean' subset of the LibriSpeech corpus [15] was used as the basis of this corpus. They are recordings without little to no noise or perturbations, which makes them an appropriate point of reference from which the corpus can be based on.

The following perturbations were added to synthesize the evaluation corpus, each with its own different varying scales:

- Reverberation
- Signal-to-noise ratio
- Number of interferences
- Input signal-to-interference ratio

These perturbations were each added sequentially to various recordings of the 'dev-clean' subset of LibriSpeech to synthesize recordings, which in turn conform the evaluation corpus. These perturbations were added using their respective implementations inside the Speechbrain framework [29]. These implementations have been empirically reviewed and used by a considerable part of the speech enhancement community, which provides certainty of their efficacy. Although these implementations are mainly used to created the training data on the fly when using Speechbrain to train speech enhancement models, for the purposes of this work, they were used to perturb offline recordings that emulate speech in real-life scenarios.

3.2.1. Reverberation Scale

The *AddReverb* function of the Speechbrain framework was used to add a reverberation effect to the recording. This implementation requires a set of room impulse responses (RIRs). The Speechbrain framework automatically uses the RIRs described in [30]. This set of RIRs is conformed from various sub-sets, and in this work the 'simulated large room' sub-set was used. It was chosen because it was the one that provided a more perceptual sense of reverberation from several subjective listening sessions.

Additionally, the *AddReverb* function can "scale" the amount of reverberation added by providing a scale factor (*rir_scale_factor*), which is a number from 0 and up. 0 represents no reverberation added; 1, one scale of reverberation; 2, two scales; etc. From subjective listening sessions, a scale factor of 3 was considered to be a "high level of reverberation", and established to be the high limit of reverberation added. This scale factor is referred here as 'reverberation scale'.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 March 2023 doi:10.20944/preprints202303.0158.v1

3.2.2. Signal-to-Noise Ratio

The *AddNoise* function of the Speechbrain framework was used to add white noise to the recording. Although this function can add different types of noises (given a set of recordings of such noises), it was decided to only add white noise since the amount of energy can be easily controlled. In this regard, the *AddNoise* function receives as an argument the signal-to-noise ratio (SNR) of the added noise. However, this function receives a lower and upper limit, and adds it in a random manner.

To have better control of how much noise is being added, both the lower and upper SNR limit were set to the same value. This SNR value was progressively increased from an established lower limit to an established upper limit. In this manner, the amount of recordings with the same SNR value will be the same. The established lower limit was set at -10 dB, which represents the very challenging scenario of having more noise than clean speech. The established upper limit was set at 30 dB, which represents the trivial scenario of imperceptible noise.

3.2.3. Number of Interferences

The *AddBabble* function of the Speechbrain framework was used to add interferences (other speakers) to the recording as additional audio streams (also known as 'babble'). The interferences were recordings of other speakers from the same dev-clean subset of LibriSpeech. To have better control on what recordings have babble added onto them, the probability of a recording having babble (*mix_prob*) was set to 100% if such recording was meant to have babble added; if not, the *AddBabble* function was not run for such recording. The lower limit of number of interferences was set to 0 and 3 as the upper limit.

3.2.4. Input Signal-to-Interference Ratio

In addition to adding interferences (or 'babble'), the amount of babble energy (here referred to as 'input SIR') was also controlled as an argument of the *AddBabble* function of the Speechbrain framework. The same method of establishing lower and upper limits of SNR values described Section 3.2.2 was used to establish the input SIR limits.

To avoid audio clipping, the *AddBabble* function applies the SIR value using the energy of the whole sum of interference streams, not the energy of each interference stream. This results in a small "perceived" increase in SIR the more interferences are added, since the energy of each interference stream is somewhat "diluted" by the number of interferences.

3.3. Evaluation Corpus

Table 1 presents all the evaluated variables and all their specific values.

Table 1. Evaluated variables and their values.

Evaluated Variable	Values
Input Length (samples)	1024, 2048, 4096, 8192, 16384, 32768, 65536, 131072, Full recording
Reverberation Scale	0, 1, 2, 3
Signal-to-Noise Ratio (dB)	-10, -5, 0, 5, 10, 15, 20, 25, 30
Number of Interferences	0, 1, 2, 3
Input Signal-to-Interference Ratio (dB)	-10, -5, 0, 5, 10, 15, 20, 25, 30, 100[†]

[†] When the number of interferences is 0, the Input SIR is set to 100, but only symbolically, since it is not used.

Each combination of values in Table 1 is referred here as an 'evaluation configuration'. Considering all the different values, this resulted in 11,664 possible evaluation configurations. However, in terms of synthesized recordings, the input length is varied during the evaluation, and not part of the perturbations that were applied to synthesize recordings. So this resulted in 1,296 'perturbation configurations'.

The 'dev-clean' subset of the LibriSpeech corpus is conformed by 40 speakers, with between 20 and 50 recordings for each speaker. From all these clean recordings, 100 were

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 March 2023 doi:10.20944/preprints202303.0158.v1

randomly selected and all the perturbation configurations were applied to them. This resulted in 129,600 synthesized recordings with which the techniques are to be evaluated. This means that for every evaluation configuration, taken from the combination of evaluated variable values in Table 1, there are 100 different distinct recordings from which evaluation metrics are to be calculated. For repeatibility, the scripts used to synthesize these recordings are freely accessible at https://github.com/balkce/seeval.

All recordings are sampled at a sampling frequency (f_s) of 16 kHz. This was inherited from the LibriSpeech corpus, but, coincidentally, it is the same sample rate with which all three speech enhancement techniques were trained.

Finally, for each synthesized recording, an accompanying text file was created to register the file path of the recording of the clean target speech, as well as the file paths of the recordings of all the interferences that are part of the added babble.

3.4. Evaluation Metrics

As explained in the last section, all three speech enhancement techniques were applied to each evaluation configuration, with 100 different recordings. This resulted in having 100 results per evaluation configuration, from which the following three metrics were recorded:

- Output signal-to-interference ratio
- Inference response time and real-time factor
- Memory usage

For repeatability, the scripts used to run the whole ealuation and record all the results are freely accessible at https://github.com/balkce/seeval.

3.4.1. Output Signal-to-Interference Ratio

The output signal-to-interference ratio can be calculated as presented in Equation 1.

$$SIR = \frac{||s_T||^2}{\sum_i^I ||s_i||^2}, \tag{1}$$

where $||.||^2$ is the norm operation used to calculate the energy from a given signal, s_T is the target signal, s_i is the ith interference, and I is the number of interferences.

As it can be deduced, to synthesize a recording with a given SIR is trivial, since the target signal and the interferences are separated from the start. However, calculating the output SIR is not as trivial, since the energy of the clean target speech and of the interferences need to extracted from the estimated target output.

A popular method, known as *bss_eval_sources* [31], is usually used to calculate the output SIR. It requires to be fed all the clean versions of the speech signals inside the mixture, including the interferences, which are available in the accompanying text files for each recording. Unfortunately, it also requires to be fed all their estimated versions as well. Because of the nature of speech enhancement, only the estimated target speech is provided; the interferences are not estimated. Fortunately, *bss_eval_sources* calculates another metric, referred to as 'signal-to-distortion ratio' (SDR), and calculates it such that (in an admittedly over-simplified manner) anything that is not the target signal is considered as a "distortion". Thus, if *bss_eval_sources* is fed only the clean target speech and the estimated target speech, the SDR is equivalent to the SIR presented in Equation 1. The implementation of *bss_eval_sources* that was used is part of the *mir_eval* package [32].

An output SIR of 20 dB can be considered of "high" enhancement performance.

3.4.2. Inference Response Time and Real-Time Factor

The response time was measured just for the inference part of the speech enhancement technique (no re-sampling or file loading was included as part of the response time). Because the synthesized recording is split into several audio segments, each of a given input length (N_w) as part of the evaluation configuration, several inferences are carried

 doi:10.20944/preprints202303.0158.v1

out for each evaluation configuration (one for each audio segment). Thus, the recorded response time is the average of the inference response times of all the fed audio segments.

It is important to mention that response times are highly dependent on the specifications of the machine that is used to carry out the evaluation. In this case, the evaluation machine had an AMD Ryzen 9 3900X 12-Core Processor; no other processing unit (like a GPU with CUDA) was used as part of the inference of the speech enhancement techniques. The reason why this was chosen was because it was of interest to see how well will the techniques respond without a GPU to handle the inference, but with an above-average CPU. This scenario is one that an enthusiast user may find themselves in, and establishes a base of reference for the response time that they can expect in their own computers.

Additionally, the real-time factor (RTF) was calculated was recorded. The RTF measures how viable a signal processing technique is to be able to run in an online manner, and is calculated as shown in Equation 2:

$$RTF = \frac{\tau_p + \tau_a}{\tau_c}, \tag{2}$$

where τ_c is the input length in seconds ($\tau_c = N_w / f_s$); τ_p, the inference response time (which was recorded); and τ_a is the sum of the response times for pre-processing and post-processing the audio segment. The latter was measured through a series of independent tests, focusing on re-sampling from the original audio sampling rate to the sampling rate of the recordings the techniques were trained with (16 kHz.), and viceversa.

If $RTF > 1$, it means that the technique is not viable to be run online and 'overruns' are to be expected. When an overrun occurs, since the audio segment was not processed in time, it is discarded and usually replaced with zeroes in the concatenation, which results in heavy distortions in the final estimation, or even in complete silence if $RTF \gg 1$. On the other hand, online viability can be implied if $RTF < 1$. However, if $RTF \lesssim 1$, one can still expect overruns to occur, but in a more sporadic fashion (since response times vary from audio segment to audio segment), thus, heavy distortions in the final estimation are to be expected. Thus, an RTF of 0.5 is usually considered preferable.

3.4.3. Memory Usage

The amount of memory the technique occupied in RAM (in bytes), at the end of the set of inferences for each evaluation configuration, was recorded. This was obtained through the assessment of the "Resident Set Size" field of the built-in memory management system in Debian Linux, reported by the *psutil* package.

4. Results

The impact of each evaluated variable is to be inspected, in respect to the input length evaluated variable. Meaning, most of the figures in this section have as the horizontal axis the varying input lengths.

For full transparency, the whole set of results are provided as supplementary material to this work. However, for simplicity, when assessing each variable, the other variables are set to their default value shown in Table 2, unless specified otherwise.

Table 2. The default values for all evaluated variables, except input length.

Variable	Default Value
Reverberation Scale	0
Signal-to-Noise Ratio (dB)	30
Number of Interferences	0
Input Signal-to-Interference Ratio (dB)	100[†]

[†] When the number of interferences is 0, the Input SIR is set to 100, but only symbolically, since it is not used.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 March 2023 doi:10.20944/preprints202303.0158.v1

Additionally, since 100 different recordings were synthesized for every evaluation configuration, the figures are shown as box plots, so that the distribution of the results are shown for each evaluation configuration, along with its median value.

The "box" in the box plot represents the two middle quartiles of all the measured SIR results, and the line that crosses the middle of the box is the median of all the measured SIR results. The difference of the minimum value and maximum value inside this box is known as the inter-quartile range (IQR). To this effect, the box also has two "whiskers": the bottom of the lower whisker represents the value that is $1.5 * IQR$ below the minimum value of the box; the top of the upper whisker represents the value that is $1.5 * IQR$ above the maximum value of the box. Finally, 'outliers' are values that are outside the range between the bottom of the lower whisker and the top of the upper whisker, and are plotted as points.

In the following sub-sections, the performance results for each evaluated variable are presented, and in Section 5 the insights from these results are discussed, with some recommendations for future work.

4.1. Output SIR vs. Input Length

In Figure 1, the separation performance (SIR) of all three techniques are shown, in respect with the input length.

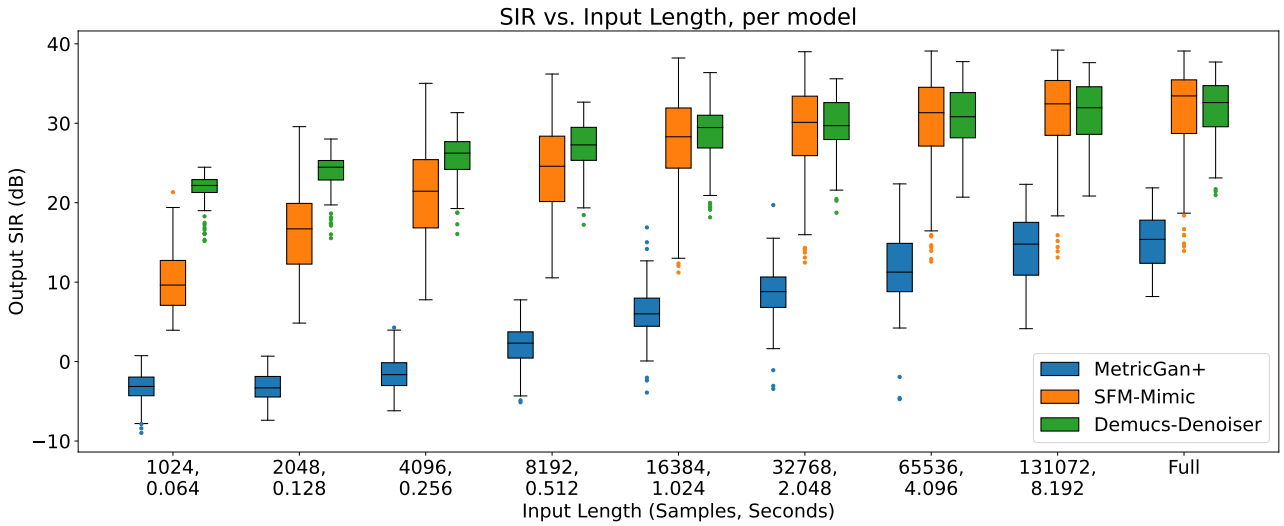

Figure 1. SIR vs input length, per model.

As it can be seen, there is a clear upper trend of performance when increasing the input length for all three techniques. However, it can be seen that Demucs-Denoiser is more robust against this change, having the best performance with small input lengths (< 4096 samples, 0.256 s.) in both terms of median SIR and SIR variance (presenting a more predictable behavior). On the other hand, MetricGan+ is outperformed by the other two techniques with every input length.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 March 2023 doi:10.20944/preprints202303.0158.v1

4.2. Output SIR vs. Reverberation Scale

In Figures 2, 3 and 4, the impact of reverberation across different input lengths are shown for MetricGan+, SFM-Mimic and Demucs-Denoiser, respectively.

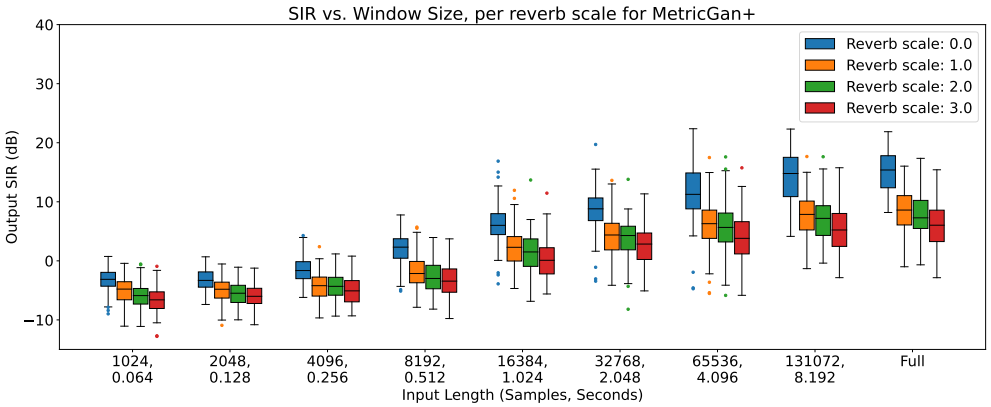

Figure 2. SIR vs input length, per reverberation scale, for MetricGan+.

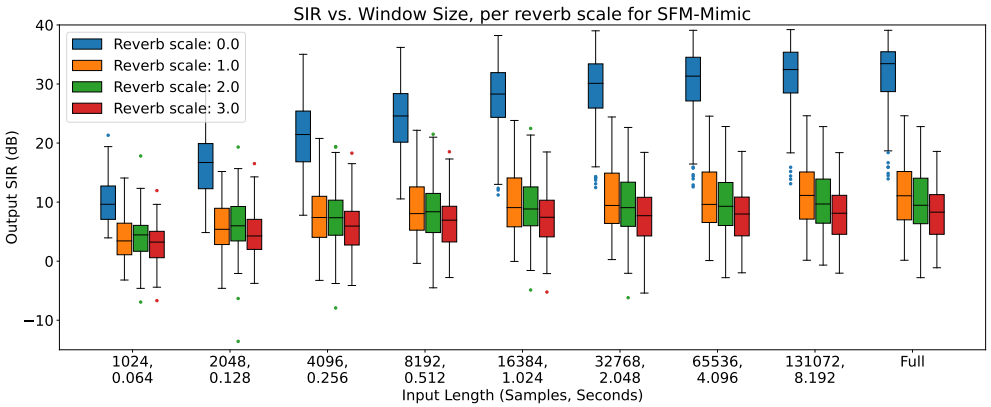

Figure 3. SIR vs input length, per reverberation scale, for SFM-Mimic.

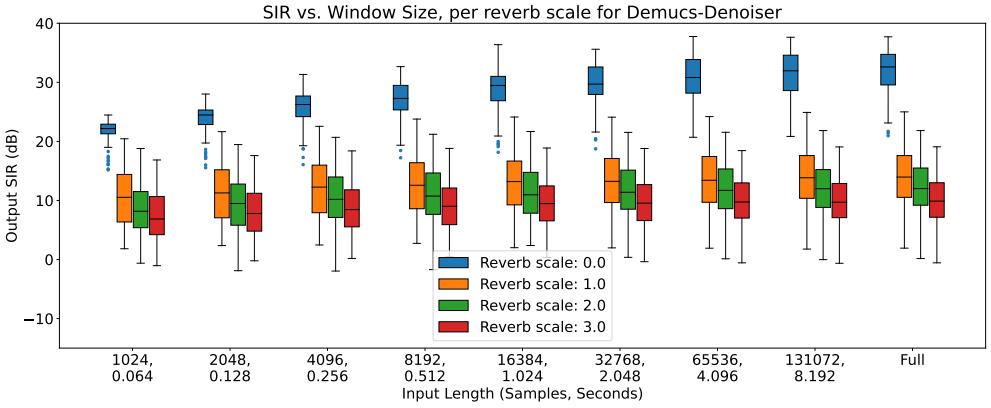

Figure 4. SIR vs input length, per reverberation scale, for Demucs-Denoiser.

As it can be seen, all models are impacted by reverberation. The input length does not seem to provide any considerable effect, except with MetricGan+ that does appear to benefit from longer input lengths (> 32768 samples, 2.048 s.).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 March 2023 doi:10.20944/preprints202303.0158.v1

4.3. Output SIR vs. Number of Interferences

In Figures 5, 6 and 7, the impact of the number of interferences across different input lengths are shown for MetricGan+, SFM-Mimic and Demucs-Denoiser, respectively. The input SIR was set to 0 dB for these results.

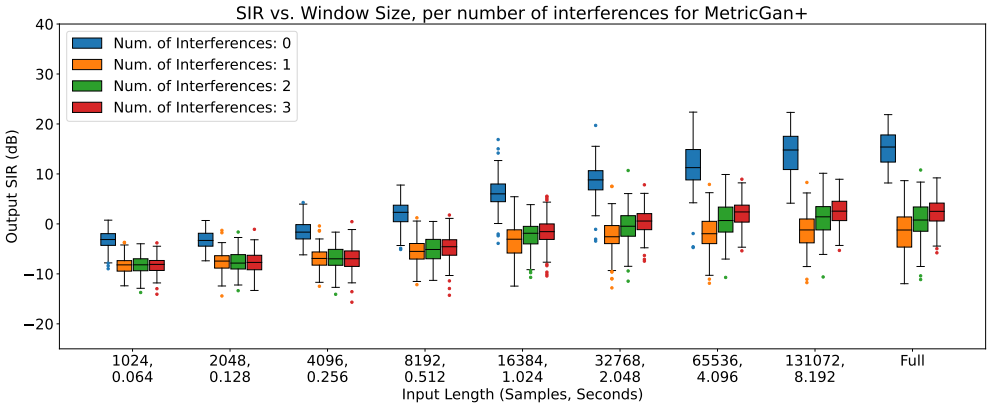

Figure 5. SIR vs input length, per number of interferences, for MetricGan+.

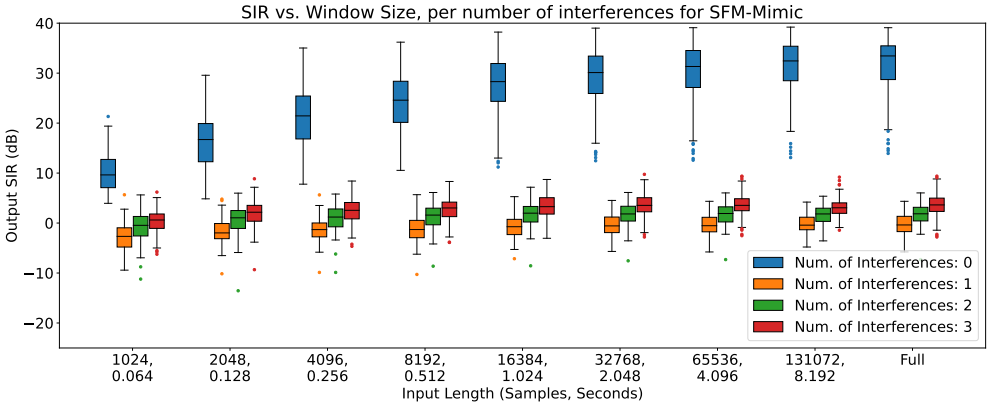

Figure 6. SIR vs input length, per number of interferences, for SFM-Mimic.

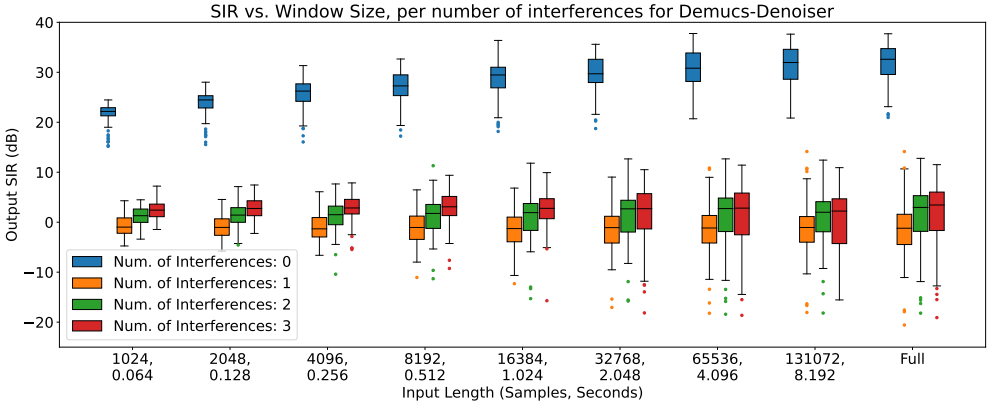

Figure 7. SIR vs input length, per number of interferences, for Demucs-Denoiser.

As it can be seen, all models are severely impacted by having interferences, with a drop of more than 10 dB in all circumstances; Demucs-Denoiser also presents a higher SIR variance, resulting in a more unpredictable behavior. Additionally, a slight upper trend can be seen when increasing the number of interferences; this could be explained by the slight increase of input SIR detailed in Section 3.2.4.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 March 2023 doi:10.20944/preprints202303.0158.v1

4.4. Output SIR vs. Input SIR

In Figures 8, 9 and 10, the impact of the input SIR across different input lengths are shown for MetricGan+, SFM-Mimic and Demucs-Denoiser, respectively. The number of interferences was set to 1 for these results.

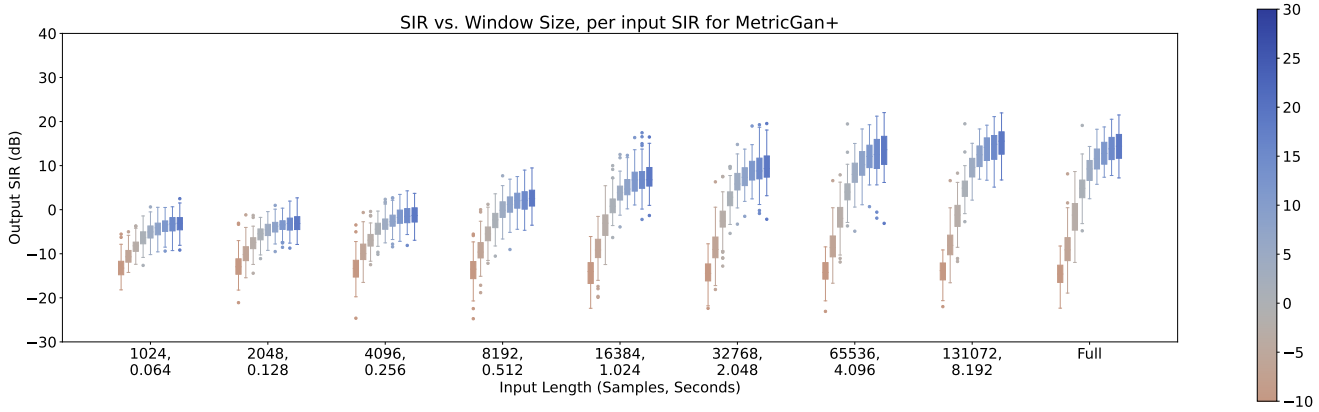

Figure 8. SIR vs input length, per input SIR, for MetricGan+.

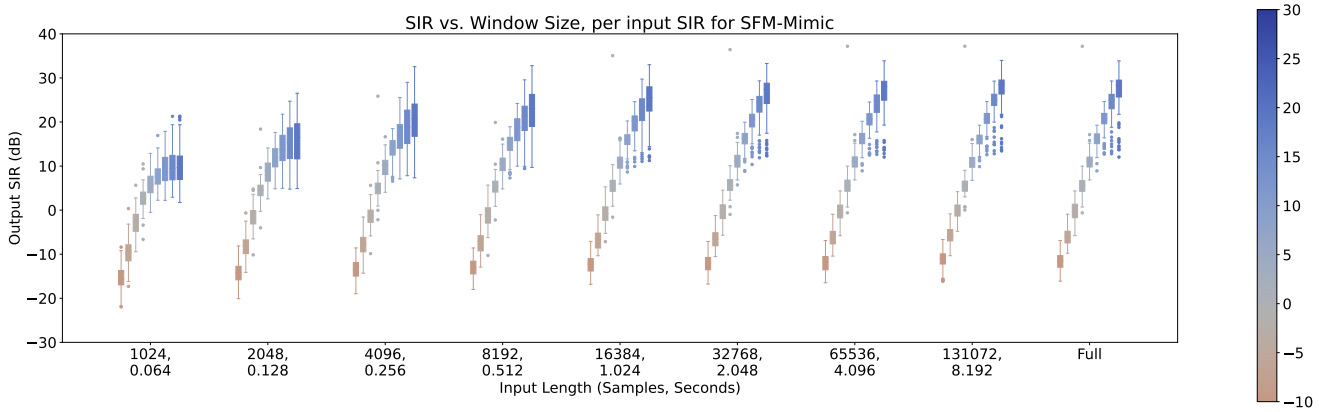

Figure 9. SIR vs input length, per input SIR, for SFM-Mimic.

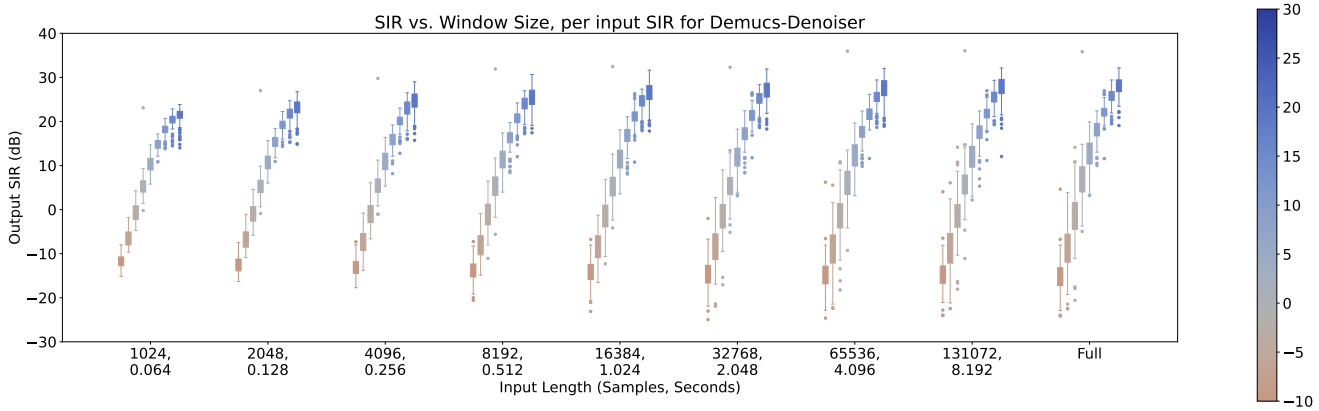

Figure 10. SIR vs input length, per input SIR, for Demucs-Denoiser.

As it can be seen, all models benefit greatly by increasing the input SIR. It is important to note that with an input SIR between -5 and 5 dB, MetricGAN+ and Demucs-Denoiser provide a large variance in their performance. This may indicate that the techniques are randomly "choosing" to estimate either the interference or the target speech when they have similar energies. This will be discussed further in Section 5.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 March 2023 doi:10.20944/preprints202303.0158.v1

4.5. Output SIR vs. SNR

In Figures 11, 12 and 13, the impact of the SNR across different input lengths are shown for MetricGan+, SFM-Mimic and Demucs-Denoiser, respectively. It's important to note that a SNR of 30 dB is virtually imperceptible.

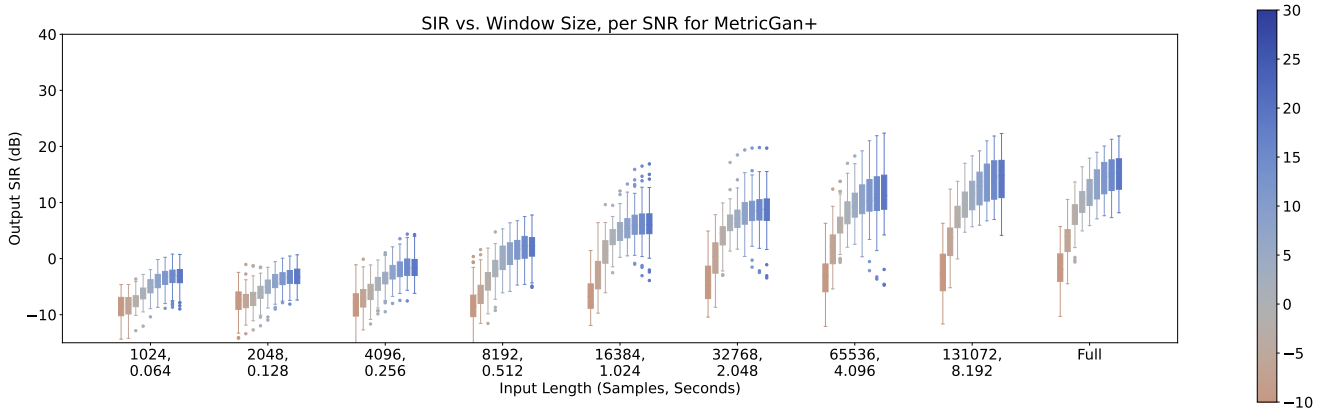

Figure 11. SIR vs input length, per SNR, for MetricGan+.

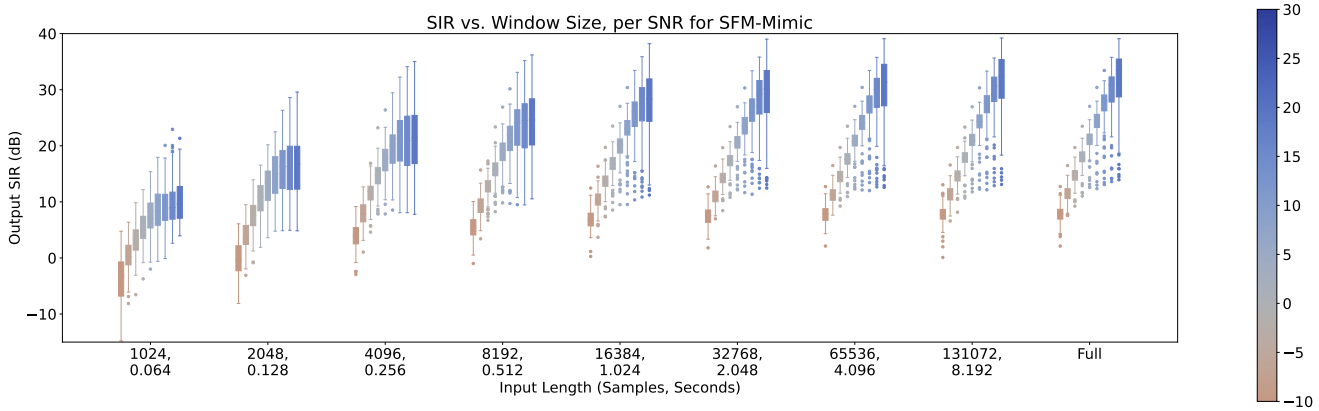

Figure 12. SIR vs input length, per SNR, for SFM-Mimic.

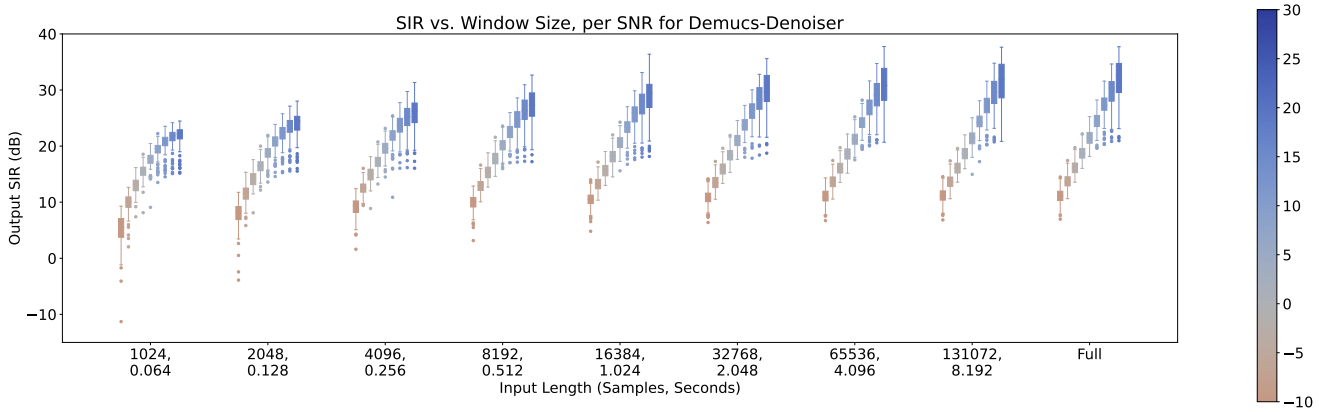

Figure 13. SIR vs input length, per SNR, for Demucs-Denoiser.

As it can be seen, all models benefit greatly by increasing the SNR. It is important to note that although Demucs-Denoiser outperforms the other two in terms of median output SIR, it still provides a similar performance as the other two techniques in the cases of very low SNR (-10 dB) and very small input lengths (< 2048 samples, 0.128 s.).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 March 2023 doi:10.20944/preprints202303.0158.v1

4.6. Real-Time Factor

The response times (τ_p) of all three techniques across different input lengths are shown in Figure 15. No response times were measured when feeding the whole recording to the techniques.

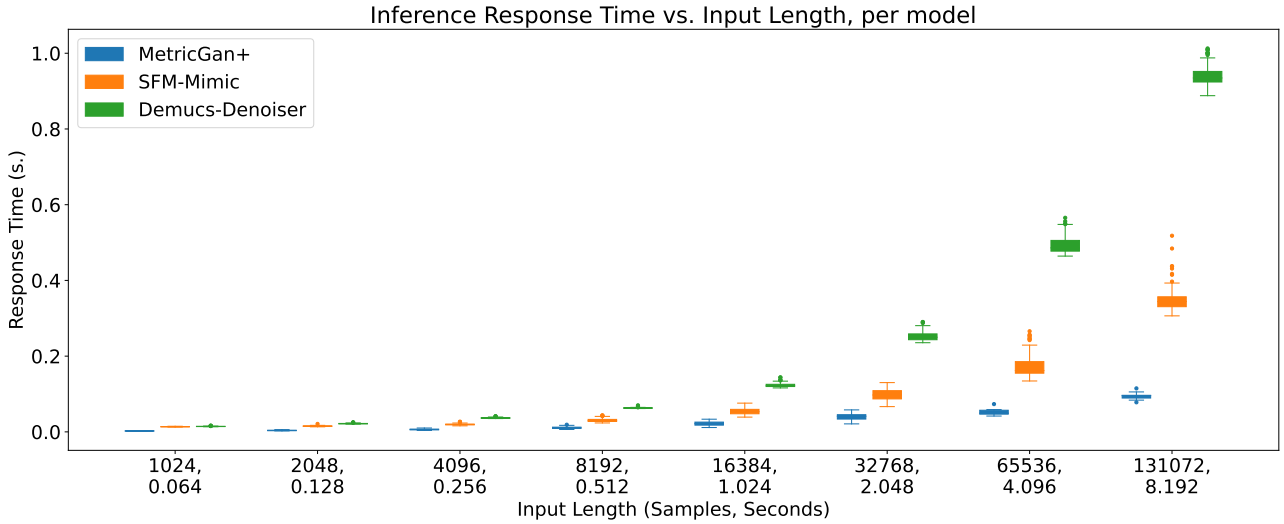

Figure 14. Response time vs input length, per model.

As detailed in Section 3.4.2, the real-time factor is calculated as $RTF = \frac{\tau_p+\tau_a}{\tau_c}$, where τ_c is the input length (in seconds) and τ_a is the sum of the response times of pre-processing and post-processing the audio segment. These were measured independently, focusing on re-sampling to and from the audio segment sampling rate, since this is the main pre- and post-processing step that is carried out during online speech enhancement. A τ_a was obtained per input length averaging the response time of 100 iterations of pre- and post-processing an audio segment of such length. The real-time factors of all three techniques across different input lengths are shown in Figure 15.

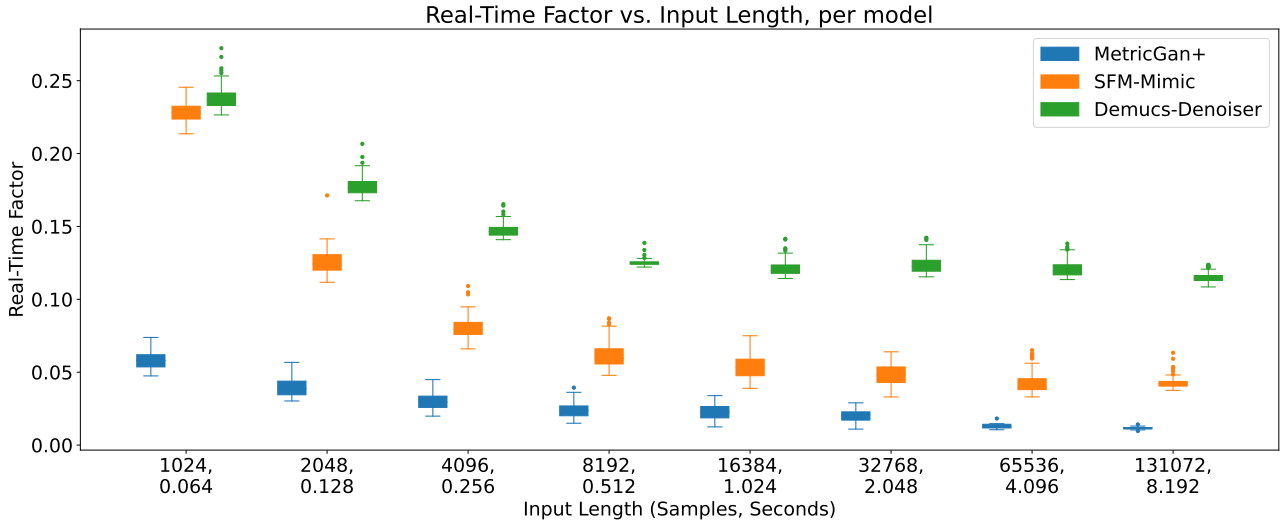

Figure 15. Real-time factor vs input length, per model.

As it can be seen, all three models decrease their real-time factors (RTFs) the longer the input lengths, even when their inference response time increase. This implies that the increase of inference response time is considerably slower than the increase of input length. As a result, their online viability increases the more information is provided, which may

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 March 2023 doi:10.20944/preprints202303.0158.v1

seem counter-intuitive, since there is more data to process; however, this also means that there is more time to process it, at a price of added latency. Furthermore, there seems to be plateau of the RTF value around the input length of 8192 samples (0.512 s.) where more information does not seem to decrease the RTFs considerably in all three techniques, with no clear impact on the variance.

Additionally, MetricGan+ provides the lowest real-time factors across all circumstances, pointing to it being the most viable to be run in an online manner. This is important since, as it can be seen in the last sections, it usually provides the lowest output SIR of all three techniques. This hints that the enhancing performance of MetricGan+ was, in a way, traded off by its quick response times.

To be fair, however, it is important to note that all three techniques have real-time factors well below its upper limit of online viability ($RTF = 1$). Meaning, that all three are viable to be run in an online manner.

4.7. Memory Usage

The memory usage (in megabytes) through the first 100 evaluation iterations of all three techniques can be seen in Figure 16. As it can be seen, MetricGAN+ starts with the lowest memory usage in those first 100 evaluation iterations.

However, an extension to 10,000 evaluation iterations is shown in Figure 17, and it can be seen that Demucs-Denoiser provides a more predictable memory usage through time. Meaning, even if it does require considerable more memory to start out than MetricGAN+, it may be preferable to it since MetricGAN+ (along with SFM-Mimic) will start increasing their memory usage substantially.

In any case, it is important to note that, even in the best case scenario, all three models require more than 500 MB of memory to carry out speech enhancement. In the worst case scenario, the memory usage for MetricGAN+ and SFM-Mimic requires to unload the model from memory and reloaded again so that memory usage stays within a practical range.

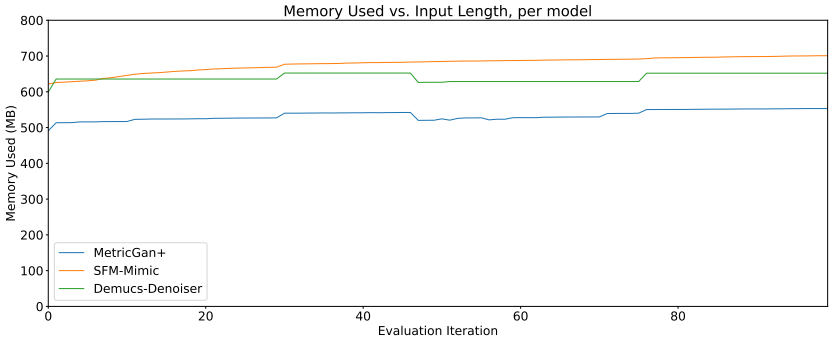

Figure 16. Memory usage through 100 evaluation iterations, per model.

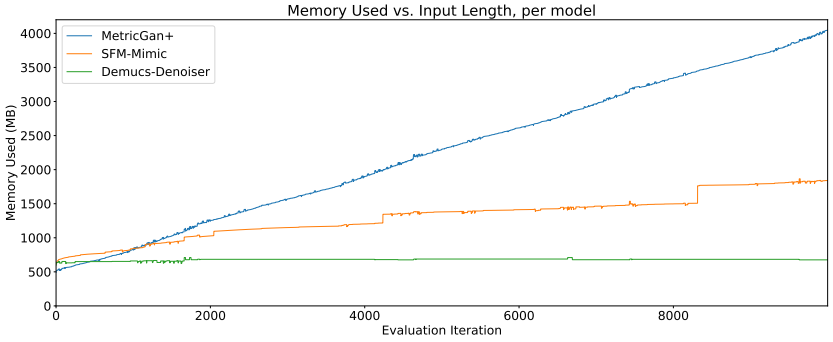

Figure 17. Memory usage through 10,000 evaluation iterations, per model.

 doi:10.20944/preprints202303.0158.v1

5. Discussion

As expected, increasing SNR and/or input SIR and decreasing the reverberation, benefits performance across all input lengths. It could be argued that with small input lengths (< 4096 samples, 0.256 s.), performance doesn't seem to be as impacted by the SNR, input SIR and reverberation as in other cases. However, because the performances in these cases are already low to begin with, the real reason is that there is a type of output SIR floor that the techniques (specially MetricGAN+ and SFM-Mimic) are hitting, and disturbing the signal further (with interferences, noise and reverberation) will not provide a worse performance.

The presence of interferences (even just one) impacts all three techniques, regardless of the input length. In the case of Demucs-Denoiser, which performs well even with the smallest input length, adding interferences impacts it the most, to the point of performing similarly as the other two techniques, which it usually outperforms in similar input length scenarios. It is important to remember that speech enhancement techniques aim to "separate" the target speech stream from the audio signal, assuming that only one speech stream is present in the audio signal. If there are more than one speech stream in the audio signal, the techniques will either: a) implicitly "decide" which one to focus on based on unforeseen biases present in the training data, usually separating the loudest one; or b) mixing all of the speech streams into one, and separating it from the rest of the audio signal. This explains the severe impact on their performances when adding interferences.

Furthermore, there is an increase in variance in performance when increasing the SNR for MetricGAN+ and for SFM-Mimic, for all input lengths. The SNR values where this phenomenon occurs are moderately high (> 10 dB) which indicates that these techniques, when they fail, they appear to do so in a more predictable manner than when they succeed. This is actually a good thing since it provides a clear point from which to refine them. As for Demucs-Denoiser, although this SNR-to-performance-variance still presents itself, it does so at a much lower rate, which points that it is much more predictable (when both failing and succeeding) across the SNR value range.

There is also an increase in performance variance when increasing the SIR for MetricGAN+ with all input lengths and for SFM-Mimic with input lengths smaller than 32768 samples (2.048 s.). Again, the SIR values where this phenomenon occurs are moderately high (> 10 dB), which re-inforces the intuition provided earlier, although its appearance it's more subtle. Interestingly, MetricGAN+ and Demucs-Denoiser tend to have high performance variance when input SIR values are between -5 and 5 dB, that is to say, when the target speech and the interferences have similar energies. As mentioned before, this may indicate that both techniques are "choosing" randomly which speech to enhance (either the target or interference), hinting that some sort of selection mechanism may be needed for these scenarios.

One could also argue that MetricGAN+ should be avoided, considering that it is being constantly outperformed by the other two techniques, and that it is the most impacted by small input lengths (< 4096 samples, 0.256 s.). However, as mentioned earlier, it provides the best real-time factor of all three, which may hint that such a dip in performance was the cost of its small response times. This is not to say that MetricGAN+ is "efficient", since it (along with SFM-Mimic) have a clear problem with their memory usage through time. The author went to great lengths to ensure that there weren't any memory leaks in the scripts used in the evaluation process. This is further evidenced by the fact that Demucs-Denoiser did not suffer from this upward trending memory consumption, while using the same evaluation scripts. Thus, it can be concluded that both MetricGAN+ and SFM-Mimic are in dire need of memory usage tuning.

All three techniques are viable to be run in an online manner, providing better results (both in terms of output SIR and RTF) the greater the input lengths. However, it is important to note that increasing the input length will result in added latency for the user. To this effect, the author recommends an input length between 4096 and 8192 samples (between 0.256 s. and 0.512 s.), for SFM-Mimic and Demucs-Denoiser. The reasoning behind this

 doi:10.20944/preprints202303.0158.v1

recommendation is that the impact in output SIR of input SIR, SNR and reverberation seem to plateau around these lengths, while providing an RTF value that is close to their respective lowest, along with a latency that could be tolerated by the average user.

6. Conclusions

Speech enhancement performance has been greatly increased by the employment of deep-learning-based techniques. Carrying out these techniques in an online manner is of great interest, but few systematic evaluations for these scenarios have been carried out. In this work, a thorough characterization of three popular speech enhancement techniques (MetricGAN+, SFM-Mimic, and Demucs-Denoiser) was carried out by emulating an online audio processing scheme. It involved sequentially feeding audio segments (of varying input lengths) of synthesized recordings to the techniques, and concatenating the results. Synthesized recordings were created from 100 randomly selected recordings of the 'dev-clean' subset of the LibriSpeech corpus, which were then disturbed by adding varying amounts of noise, varying amounts of interferences with varying amounts of energy, and varying amounts of reverberation, resulting in a sizable evaluation corpus. The scripts to create this evaluation corpus, as well as to carry out the systematic characterization, are freely accessible at https://github.com/balkce/seeval.

It was found that small window lengths (< 4096 samples, 0.256 s.) are detrimental to the enhancement performance (measured as output SIR). However, Demucs-Denoiser was the least affected by it. Furthermore, this technique provided the least result variance when increasing SNR, providing predictable results, although it was affected overall similarly as SFM-Mimic. On the other hand, MetricGAN+ was consistently outperformed by the other two techniques, but it was observed that this could be considered as a trade off for its low response times. However, all three techniques provided real-time factor values well below the threshold for online viability, with the author recommending lengths between 4096 and 8192 samples (between 0.256 s. and 0.512 s.) for SFM-Mimic and Demucs-Denoiser, since they provide a balance between output SIR (against all disturbances), RTF and latency.

All three models were severely impacted by reverberation, for all input lengths. They were also severely impacted by the presence of interferences; with even just one was enough to see dips in performance of more than 10 dB. In terms of SIR, although they were all impacted as expected, Demucs-Denoiser showed an increase in result variance when the target speech and the interference had a similar energy (input SIR between -5 and 5 dB). This hints at the possibility of "confusion" of which speech stream to "focus on", resulting in the techniques randomly choosing to separate the interference instead of the target speech. It is proposed, as future work, to implement a type of selection mechanism for Demucs-Denoiser so that this issue is resolved.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/1010000/s1, Spreadsheet S1: Full Results for MetricGAN+; https://www.mdpi.com/article/10.3390/1010000/s2, Spreadsheet S2: Full Results for SFM-Mimic; https://www.mdpi.com/article/10.3390/1010000/s3, Spreadsheet S3: Full Results for Demucs-Denoiser.

Funding: This work was supported by PAPIIT-UNAM through the grant IA100120.

Data Availability Statement: All of the scripts employed to carry out the characterization presented in this work are publicly accessible in https://github.com/balkce/seeval.

Acknowledgments: The author would like to thank Juan Pablo Maldonado-Castro and Felix Sanchez Morales since their insights in their own independent projects (Bachelor and Masters, respectively) inspired this work.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Das, N.; Chakraborty, S.; Chaki, J.; Padhy, N.; Dey, N. Fundamentals, present and future perspectives of speech enhancement. *International Journal of Speech Technology* **2021**, *24*, 883–901.

2. Wang, D.; Chen, J. Supervised Speech Separation Based on Deep Learning: An Overview. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **2018**, *26*, 1702–1726. https://doi.org/10.1109/TASLP.2018.2842159.
3. Eskimez, S.E.; Soufleris, P.; Duan, Z.; Heinzelman, W. Front-end speech enhancement for commercial speaker verification systems. *Speech Communication* **2018**, *99*, 101–113.
4. porov, a.; oh, e.; choo, k.; sung, h.; jeong, j.; osipov, k.; francois, h. music enhancement by a novel cnn architecture. *journal of the audio engineering society* **2018**.
5. Lopatka, K.; Czyzewski, A.; Kostek, B. Improving listeners' experience for movie playback through enhancing dialogue clarity in soundtracks. *Digital Signal Processing* **2016**, *48*, 40–49.
6. Li, C.; Shi, J.; Zhang, W.; Subramanian, A.S.; Chang, X.; Kamo, N.; Hira, M.; Hayashi, T.; Boeddeker, C.; Chen, Z.; et al. ESPnet-SE: End-to-end speech enhancement and separation toolkit designed for ASR integration. In Proceedings of the 2021 IEEE Spoken Language Technology Workshop (SLT). IEEE, 2021, pp. 785–792.
7. Rascon, C.; Meza, I. Localization of sound sources in robotics: A review. *Robotics and Autonomous Systems* **2017**, *96*, 184–210.
8. Lai, Y.H.; Zheng, W.Z. Multi-objective learning based speech enhancement method to increase speech quality and intelligibility for hearing aid device users. *Biomedical Signal Processing and Control* **2019**, *48*, 35–45.
9. Zhang, Q.; Wang, D.; Zhao, R.; Yu, Y.; Shen, J. Sensing to hear: Speech enhancement for mobile devices using acoustic signals. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* **2021**, *5*, 1–30.
10. Rao, W.; Fu, Y.; Hu, Y.; Xu, X.; Jv, Y.; Han, J.; Jiang, Z.; Xie, L.; Wang, Y.; Watanabe, S.; et al. Conferencingspeech challenge: Towards far-field multi-channel speech enhancement for video conferencing. In Proceedings of the 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). IEEE, 2021, pp. 679–686.
11. Hu, Y.; Loizou, P.C. Subjective comparison and evaluation of speech enhancement algorithms. *Speech communication* **2007**, *49*, 588–601.
12. Upadhyay, N.; Karmakar, A. Speech enhancement using spectral subtraction-type algorithms: A comparison and simulation study. *Procedia Computer Science* **2015**, *54*, 574–584.
13. Nossier, S.A.; Wall, J.; Moniri, M.; Glackin, C.; Cannings, N. A comparative study of time and frequency domain approaches to deep learning based speech enhancement. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN). IEEE, 2020, pp. 1–8.
14. Graetzer, S.; Hopkins, C. Comparison of ideal mask-based speech enhancement algorithms for speech mixed with white noise at low mixture signal-to-noise ratios. *The Journal of the Acoustical Society of America* **2022**, *152*, 3458–3470.
15. Panayotov, V.; Chen, G.; Povey, D.; Khudanpur, S. Librispeech: An ASR corpus based on public domain audio books. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015, pp. 5206–5210. https://doi.org/10.1109/ICASSP.2015.7178964.
16. Bagchi, D.; Plantinga, P.; Stiff, A.; Fosler-Lussier, E. Spectral Feature Mapping with MIMIC Loss for Robust Speech Recognition. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 5609–5613. https://doi.org/10.1109/ICASSP.2018.8462622.
17. Fu, S.W.; Yu, C.; Hsieh, T.A.; Plantinga, P.; Ravanelli, M.; Lu, X.; Tsao, Y. MetricGAN+: An Improved Version of MetricGAN for Speech Enhancement. In Proceedings of the Proc. Interspeech 2021, 2021, pp. 201–205. https://doi.org/10.21437/Interspeech.2021-599.
18. Défossez, A.; Synnaeve, G.; Adi, Y. Real Time Speech Enhancement in the Waveform Domain. In Proceedings of the Proc. Interspeech 2020, 2020, pp. 3291–3295. https://doi.org/10.21437/Interspeech.2020-2409.
19. Hao, X.; Xu, C.; Xie, L.; Li, H. Optimizing the Perceptual Quality of Time-Domain Speech Enhancement with Reinforcement Learning. *Tsinghua Science and Technology* **2022**, *27*, 939–947. https://doi.org/10.26599/TST.2021.9010048.
20. Zeng, Y.; Konan, J.; Han, S.; Bick, D.; Yang, M.; Kumar, A.; Watanabe, S.; Raj, B. TAPLoss: A Temporal Acoustic Parameter Loss for Speech Enhancement. *arXiv preprint arXiv:2302.08088* **2023**.
21. Jain, S.M., Hugging Face. In *Introduction to Transformers for NLP: With the Hugging Face Library and Models to Solve Problems*; Apress: Berkeley, CA, 2022; p. 51–67. https://doi.org/10.1007/978-1-4842-8844-3_4.
22. Creswell, A.; White, T.; Dumoulin, V.; Arulkumaran, K.; Sengupta, B.; Bharath, A.A. Generative Adversarial Networks: An Overview. *IEEE Signal Processing Magazine* **2018**, *35*, 53–65. https://doi.org/10.1109/MSP.2017.2765202.
23. Fu, S.W.; Liao, C.F.; Tsao, Y.; Lin, S.D. MetricGAN: Generative Adversarial Networks based Black-box Metric Scores Optimization for Speech Enhancement. In Proceedings of the Proceedings of the 36th International Conference on Machine Learning; Chaudhuri, K.; Salakhutdinov, R., Eds. PMLR, 2019, Vol. 97, *Proceedings of Machine Learning Research*, pp. 2031–2041.
24. Recommendation, I.T. Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs. *Rec. ITU-T P. 862* **2001**.
25. Taal, C.H.; Hendriks, R.C.; Heusdens, R.; Jensen, J. A short-time objective intelligibility measure for time-frequency weighted noisy speech. In Proceedings of the 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, 2010, pp. 4214–4217. https://doi.org/10.1109/ICASSP.2010.5495701.
26. Défossez, A.; Usunier, N.; Bottou, L.; Bach, F. Music source separation in the waveform domain. *arXiv preprint arXiv:1911.13254* **2019**.

 doi:10.20944/preprints202303.0158.v1

27. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015; Navab, N.; Hornegger, J.; Wells, W.M.; Frangi, A.F., Eds.; Springer International Publishing: Cham, 2015; p. 234–241.
28. Reddy, C.K.; Gopal, V.; Cutler, R.; Beyrami, E.; Cheng, R.; Dubey, H.; Matusevych, S.; Aichner, R.; Aazami, A.; Braun, S.; et al. The interspeech 2020 deep noise suppression challenge: Datasets, subjective testing framework, and challenge results. *arXiv preprint arXiv:2005.13981* **2020**.
29. Ravanelli, M.; Parcollet, T.; Plantinga, P.; Rouhe, A.; Cornell, S.; Lugosch, L.; Subakan, C.; Dawalatabad, N.; Heba, A.; Zhong, J.; et al. SpeechBrain: A General-Purpose Speech Toolkit, 2021, [arXiv:eess.AS/2106.04624]. arXiv:2106.04624.
30. Ko, T.; Peddinti, V.; Povey, D.; Seltzer, M.L.; Khudanpur, S. A study on data augmentation of reverberant speech for robust speech recognition. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2017, pp. 5220–5224.
31. Vincent, E.; Gribonval, R.; Fevotte, C. Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing* **2006**, *14*, 1462–1469. https://doi.org/10.1109/TSA.2005.858005.
32. Raffel, C.; McFee, B.; Humphrey, E.J.; Salamon, J.; Nieto, O.; Liang, D.; Ellis, D.P.; Raffel, C.C. MIR_EVAL: A Transparent Implementation of Common MIR Metrics. In Proceedings of the ISMIR, 2014, pp. 367–372.