ShallowFake: Adversarial Attacks on Speech Recognition and Audio Deepfakes

Praneet Rathi, Aman Khinvasara, and Shrey Patel
University of Illinois at Urbana-Champaign

Abstract

Incriminating evidence of your favorite politician admitting to corruption? Audio DeepFakes present a large attack surface, and individual privacy is consistently violated with intrusive speech recording and recognition systems. We seek to protect against such scenarios by minimally perturbing sound to preserve human comprehension while preventing successful speech recognition and DeepFake synthesis. We present adversarial Projected Gradient Descent (PGD) attacks on OpenAI's Whisper speech recognition system and SV2TTS speech synthesis system, an audio DeepFake. These results are used to train noising GANs for each system. We benchmark these approaches with statistical and deep-learning based metrics.

1 Introduction

Rapid progress in machine learning applications for everyday use cases promises massive leaps in productivity and well-being. In particular, speech recognition systems, such as OpenAI‘s recent Whisper [Radford et al. 2023] can streamline everyday consumer and enterprise interactions. Audio deepfakes such as SV2TTS [Jia et al. 2018] can create hyper-personalized experiences, bridge language barriers, and enable new forms of interacting with loved ones. Yet, today's world is rife with opportunities for adversaries to bend computational resources to their advantage, misinforming us and violating our privacy everyday, using well-intentioned tools such as speech recognition (SR) systems and audio deepfakes (DF). We must develop methods to protect individuals from these tools.

This work makes the following contributions:

1. A projected gradient descent (PGD) attack on Whisper [Radford et al. 2023], a refinement of Olivier and Raj [Olivier and Raj 2023] that is more attuned to human perception
2. A generative adversarial network (GAN) that minimally perturbs audio to attack Whisper
3. A PGD attack on an audio deepfake/voice-cloning system (SV2TTS [Jia et al. 2018])
4. A GAN trained to minimally perturb audio to attack SV2TTS
5. Comprehensive evaluation of these attacks on perception and content of original and generated samples, along with a code repository [1]

2 Adversarial Attacks

This section presents our architecture, methodology and data for each attack.

2.1 Whisper PGD Attack

Our first attack, adapted from [Olivier and Raj 2023], is a projected gradient descent attack on Whisper. PGD iteratively adds noise to the original signal before passing it to Whisper, as shown in Figure 1. It computes the gradient of the objective function with respect to the added noise, and moves to optimize the objective function. The projection step ensures that the added noise remains small relative to the original signal - in particular, we constrain the signal to noise ratio (SNR) to be greater than 35db.

The optimization is as follows:

$$\max_{\|\delta\|_p \leq \epsilon} \mathcal{L}_{\text{CE}}(f(x+\delta, y) + \lambda \mathcal{L}_{\text{STOI}}(x, x+\delta) \tag{1}$$

for audio x, correct transcription y, Whisper model f, cross-entropy loss $\mathcal{L}_{\text{CE}}$, STOI loss $\mathcal{L}_{\text{STOI}}$, loss coefficient λ, norm $p = 2$, and noise ϵ.

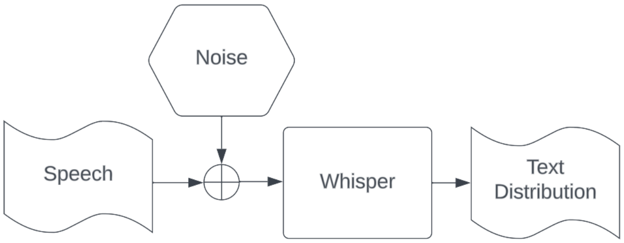

Figure 1: *Whisper PGD attack - forward pass*

2.2 Whisper GAN Attack

The iterative nature of the previous attack 2.1 presents a significant downside. In a real-time setting it is not computationally feasible to afford 200+ iterations to generate an attack. This motivates our novel use of the pix2pix architecture [Isola et al. 2017] to learn clean-to-noise wave mappings to efficiently add noise in a single inference pass. See Section 2.5 for more information about model architecture and decisions. We convert the clean and Whisper PGD-based noised pairs into spectrograms and process these to use as training data for the pix2pix model. We use log mel spectrograms and rescale to zero-mean. We use the differential noise, as opposed to clean + noise, to isolate our objective. As we seek to generalize to unseen audio sample characteristics, such as varying length, we split these spectrograms into overlapping, consistently-sized segments of approximate phrases, with an example below. The overlapping slices enable the model to see different contextual windows when predicting noise, with the aim of better generalizability. This is heuristically obtained by interpreting sharp drops in clean spectrogram magnitude as silence and splitting based on the total number of words in the audio's correct transcription. To avoid data leakage, we perform random training/validation/test split on the original samples instead of individual slices.

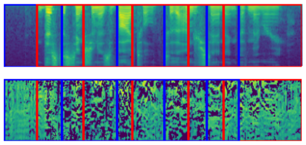

Figure 2: *Pre-processing of sample clean and noised pair. Top and bottom show clean and noised versions, and red/blue rectangles show sliced versions (identical for clean and noise).*

For audio reconstruction, we construct a unified spectrogram by, for each time slice, a weighted average between the two slices predicted for the two chunks it belongs to. The weight is proportional to its location in the overlap segment.

2.3 Deepfake PGD Attack

We adapt the Whisper PGD attack to directly attack SV2TTS [Jia et al. 2018]. SV2TTS takes a voice sample and the desired text, and

[1] https://github.com/prrathi/shallowfake

aims to output a waveform of the same speaker saying the desired text. Our dataset of voice samples is accompanied by their text label - on each iteration, we provide SV2TTS with this text label and a perturbed version of the original waveform (where the perturbation is constrained to $>$ 35 SNR). In each training pass we disconnect the vocoder from the end of the network and only train on a modified version of the network where the audio signal is passed through an encoder network and then passed into a synthesizer network to output a spectrogram of the cloned speech. We looked into maximizing a few different loss functions, (see Subsec 3.4) but found that there is a lack of robustness for speaker similarity metrics. Thus we ended up experimenting with **L1, L2** and **K-L divergence** loss. In the final objective we went with L1 loss.

We seek to optimize the following objective:

Equation 2.

$$\begin{aligned} L_1(o, \hat{o}) = \sum |o - \hat{o}| \\ f_\theta(x + \hat{x}) = \hat{y} \\ \text{argmax}_{\hat{x}}[L_1(A_1 \times \hat{y}, A_2 \times \phi(x))] \end{aligned}$$

Where $\hat{x}$ and x are the noise for the attack and the input waveform respectively. ϕ is a function that given a waveform returns the log mel-spectrogram. θ are the parameters of the network and $\hat{y}$ is the output spectrogram of our Deepfake network. A_1 and A_2 are the warping matrices generated via DTW.

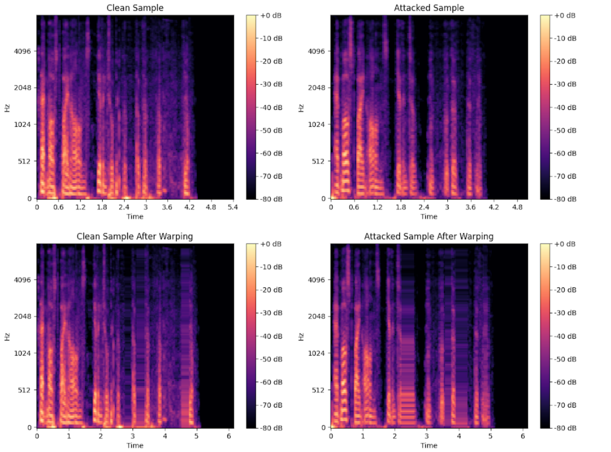

Figure 3: *Effect of dynamic time warping on noisy and clean spectrograms*

Notably, this loss function requires the generated spectrogram and the clean speech spectrogram to have the same dimension, which is not true in general. Word utterances tend to get warped after they are passed through the Deep Fake. We apply dynamic time warping on input spectogram and output spectogram to remedy this.

2.3.1 Deepfake PGD on Text

We construct a second PGD attack on SV2TTS , which attacks the text content of the generated speech. In particular, we attempted to minimize the probability that passing the generated waveform through Whisper resulted in the correct text. We tried various objective functions related to this probability, including the probability of the entire string, the average of individual logit probabilities, Whisper's internal loss function, and more. None of these attempts were successful. Even after removing the projection step from the attack, the objective function did not consistently climb over 200 iterations. The culprit is likely the additional whisper step; it may have made the objective function surface more difficult to optimize over, or may have severely dampened the effect of changes to the noise.

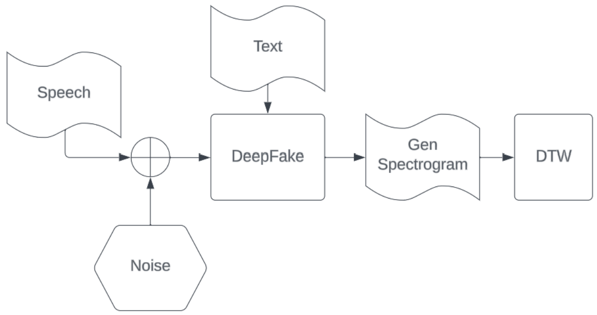

Figure 4: *Deepfake PGD attack - forward pass*

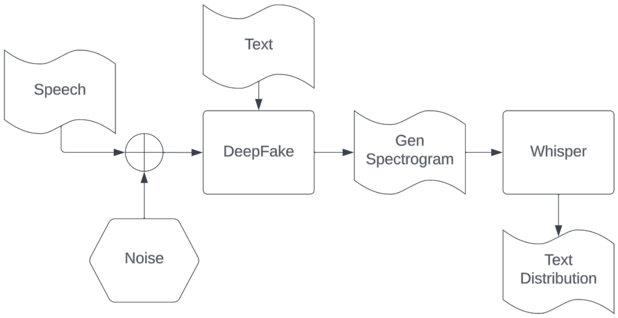

Figure 5: *Text-based Deepfake PGD attack - forward pass*

2.4 Deepfake GAN Attack

We also train a GAN to learn the noise added by the Deepfake PGD attack. See Subsection 2.2 for identical pre-processing procedure, and Subsection 2.5 for training details.

2.5 GAN Architecture

We provide further details on the Pix2Pix model we used. We adapt the model architecture from this repo. The model consists of a UNet used by the generator, outputting images the same size as the input. This has a stacked architecture with layers 16 $\rightarrow$ 32 $\rightarrow$ 64 $\rightarrow$ 128 $\rightarrow$ 256 and reversed for upsampling. The GAN's loss function consists of discriminator's probability of getting fooled, and MAE with ground truth. The discriminator trains on $\frac{1}{4}$ sized patches of the real/fake images and has 64 $\rightarrow$ 128 $\rightarrow$ 256 sized architecture. We found using Wasserstein loss [Adler and Lunz 2018] for the discriminator reduced mode collapse and increased stability during training. Since our input images are variable length due to the algorithmic splitting, we modify the architecture to be fully convolutional to allow for arbitrary data input. However, each given batch must have fixed size, so we sort all the images in each split and zero pad smaller images to the largest image in that batch. For the Whisper GAN, we use learning rate 1e-5 for the discriminator, Wasserstein weight 5, learning rate 2e-4 for the generator, and MAE coefficient 2. For the Deepfake GAN, we use discriminator learning rate 1e-5, Wasserstein weight 5, generator learning rate 1e-4, and MAE coefficient 2. Both GANs were trained for 30 total epochs.

3 Results

3.1 Dataset

We train all of our models and attacks on the LibriSpeech [Panayotov et al. 2015] ASR corpus. Specifically we use the 100 hour subset of ”clean” speech found here. Preprocessing was kept to a minimum. The data is sampled at 16 Khz and preprocessing was simply removing long silences through VAD (voice activity detection). If VAD detected silence for over a window of 1440 milliseconds that frame would be dropped from the resulting audio. The original dataset consisted of 2300 samples of length 5-15 seconds, and we used a randomly selected 200-sized subset for all evaluations. This test set was a subset of the GAN test sets, to prevent leakage.

3.2 Methodology

For each attack, we seek to understand the effect of the noising both in human perception and on the ability to generate a Deepfake with it. The attacks on Whisper are thus transfer attacks, for the ultimate goal of preventing successful deepfaking by SV2TTS .

For each original sample, for each attack, we therefore have three pieces of information to compare: (1) **clean** waveform; (2) **perturbed** version created by attack; and (3) **generated** waveform synthesized by SV2TTS , with the noisy version and original text label as input. We then compute the 3 following metrics:

1. stoi(**clean**, **perturbed**) (Short-Time Objective Intelligibility (STOI) [Taal et al. 2010]) metric measures the intelligibility of human speech. Here, we capture the human comprehension similarity between the clean version and the perturbed version. When an individual is speaking, their audience should be able to hear and understand them very similarly, while Deepfakes should not work.

2. spectralCompare(**clean**, **generated**) measures how similar the generated speaker sounds to the original speaker? This is really the l1Entropy of the spectrograms, after passing them through DTW. Because DTW will push them to be similar, this is a conservative analysis of our methods.

3. textCompare(**clean text**, **generated**) uses Whisper to measure how closely the generated waveform matches the text passed to SV2TTS during generation. This metric is really a wrapper around Whisper’s loss function, normalized by how closely the clean waveform matches the text label.

The PGD attacks were all run for 200 iterations, and all measurements come from the test set of 300 audio samples. For Whisper PGD we set $\lambda = 100$. The effectiveness of the original Whisper attack in terms of Word Error Rate over the iterations can be found in Figure 8.

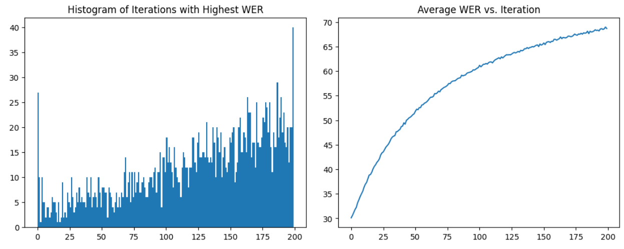

Figure 6: *Success of Whisper PGD attack, measured in Word Error Rate (WER). First shows distribution of iteration maximizing WER, and second shows average WER over iterations.*

3.3 Stoi

Figure 7 presents the intelligibility comparison of the clean and noisy signal after each attack. As expected from the projection step, we see that the stoi value is close to 1 for all attacks. Whisper PGD erodes the intelligibility the least, and the GAN trained on it *mostly* follows this behavior, although with more leftward skew. The DF PGD attack surprisingly has more leftward skew, although it remains minimal - this may be explained by the extra levels added by the DF. The tight STOI numbers are optimistic for all attacks.

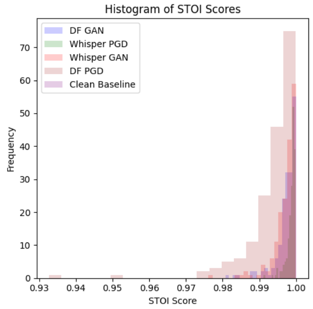

Figure 7: *Stoi comparison of clean and noisy signals across attacks*

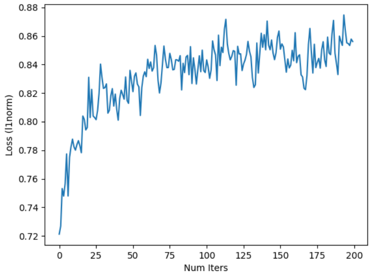

Figure 8: *Average spectrogram L1 loss per sample at ith iteration*

3.4 Spectral Comparison

We would like to attack SV2TTS on either (or both) of two dimensions: 1) speaker identity and 2) information content. We use spectral comparison as a proxy for speaker identity. We see in Figure 9 that, surprisingly, the PGD attack directly on the deepfake actually provided inferior results than both of the Whisper attacks. This is an even more positive indicator towards the feasibility of transfer attacks from speech recognition systems to audio Deepfakes. Additionally, note that the values for the Whisper GAN and Whisper PGD attack are almost identical, another validation of using a quick GAN rather than an expensive iterative approach.

Table 1: *Average Metrics by Method*

Method	STOI	Spectral Entropy	Text Score
Clean	1.0	0.636	3.089
DF PGD	0.994	0.610	3.284
DF GAN	0.997	0.641	3.115
Whisper PGD	0.999	0.644	3.070
Whisper GAN	0.997	0.648	3.115

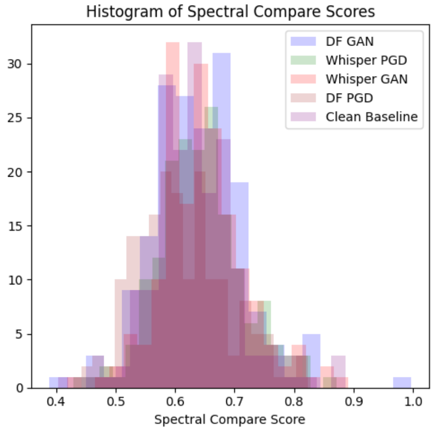

Figure 9: *Spectral similarity of SV2TTS output with clean signal*

3.5 Text Comparison

We evaluate the generated waveforms' second dimension (information content) by passing it through Whisper and measuring the likelihood of returning the correct text (using their loss function, so larger values mean that correct text is less likely). As we would expect, Figure 10 shows that performing a PGD attack directly on SV2TTS is the most effective attack. More interestingly, the non-determinism introduced by the Whisper GAN seems to moderately effective, indicating that the transfer attack might be successful. In particular, note that SV2TTS and Whisper are completely independent, making the success of the Whisper GAN in attacking SV2TTS more exciting!

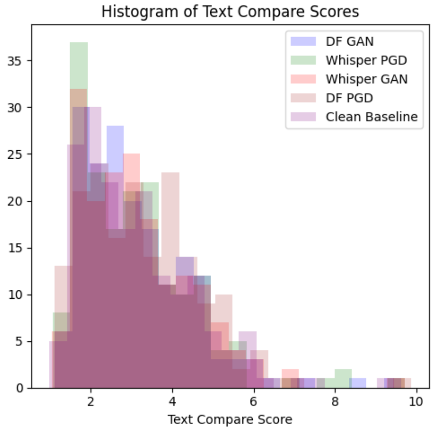

Figure 10: *Text similarity of SV2TTS output with clean text*

3.6 Head-to-Head Comparison

Figures 11, 13, and 12 show, for each metric, the number of samples for which each attack (on vertical axis) performed better than each other attack. Larger stoi values are better, showing that the perturbation is less noticable for humans. Larger spectral comparison and text comparison values are also better, indicating more divergence between the original signal and that generated by the Deepfake.

3.7 Demos

Click on the following link to access a google drive containing some attack demos. Each demo contains the attacked input sample, the clean input sample, the attacked output sample and the clean output

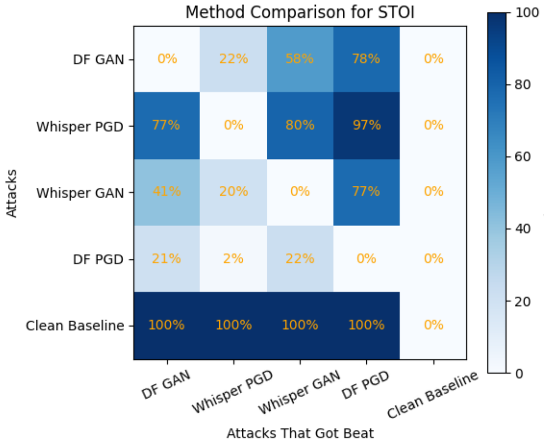

Figure 11: *Stoi comparison*

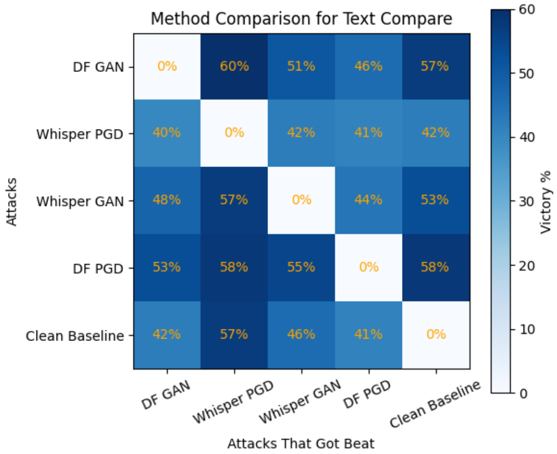

Figure 12: *Text Likelihood Comparison*

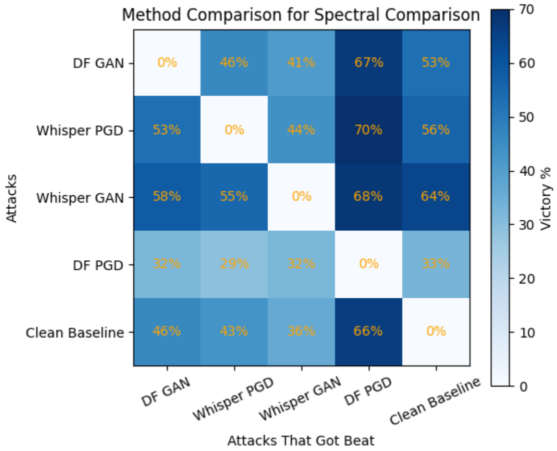

Figure 13: *Spectrogram Divergence Comparison*

sample (the latter 2 are the outputs of the deepfake)

3.8 Discussion

The proposed attack methods generally achieve larger entropy and text loss scores than the clean baseline. Simultaneously, they're able to maintain high STOI, particularly with the STOI-optimized Whisper PGD attack. Notably, the GANs perform competitively with their respective PGD attacks. One exception is Deepfake PGD, which underperforms in spectral entropy, and Whisper PGD, which underperforms in text score. We utilized a 50x decoder speed up that sacrificed precision for attack time, and suspect that this may partially contribute to the Deepfake PGD performance. We also note that training on Whisper is somewhat uncorrelated with text loss, since the latter is the output of Deepfake.

Similar trends are reflected in the Head-to-Head comparison. Of the attacks, Whisper PGD outperforms all others in STOI, Deepfake PGD outperforms in text likelihood, and Whisper GAN outperforms in spectrogram divergence.

4 Conclusion

One major outcome from our work is that we can use deterministic attacks on speech recognition systems to train a GAN to attack the same system with similar performance. Excitingly, this GAN can reasonably be used to perturb inputs to an audio Deepfake, preventing malicious voice cloning. In practice, a larger GAN could be trained on the full LibriSpeech dataset using the methods and metrics discussed in our paper. This GAN could be used to generate accurate noise to protect speaker identities by injecting noise on the fly.

Future Work Our work so far passes the original text label back to the SV2TTS for generation. In some ways, this makes the Deepfake's task easier - the voice sample it is receiving is very similar to the sound it should generate. A dataset with multiple samples from each speaker would allow more generalized experimentation.

Additionally, an open question remains as to whether these attacks work 'over-the-air,' which is particularly important for the use-case we target. The original paper attacking Whisper [Olivier and Raj 2023] found that their method didn't translate to settings where noise was added prior to a sound being transmitted by a speaker (and captured by a microphone prior to running through Whisper). We have yet to determine if any of our attacks are robust to 'over-the-air' transmission.

Somewhat orthogonally to the focus of our work, we found a gaping hole in speaker identification literature. In Section 3.4, we discussed the 'speaker identity' comparison between the clean sample and that generated by SV2TTS . We used l1Entropy as a proxy for this, but only after seeking and surprisingly failing to find a different model or metric that can answer if two waveforms (potentially saying different things) came from the same speaker. Such a system would be an exciting future contribution to the literature.

References

Adler, J., and Lunz, S. 2018. Banach wasserstein gan. *Advances in neural information processing systems 31.*

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 1125–1134.

Jia, Y., Zhang, Y., Weiss, R., Wang, Q., Shen, J., Ren, F., Nguyen, P., Pang, R., Moreno, I. L., and Wu, Y. 2018. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. In *Advances in neural information processing systems*, vol. 31.

Olivier, R., and Raj, B., 2023. There is more than one kind of robustness: Fooling whisper with adversarial examples.

Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. 2015. Librispeech: An asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5206–5210.

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., and Sutskever, I. 2023. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, 28492–28518.

Taal, C. H., Hendriks, R. C., Heusdens, R., and Jensen, J. 2010. A short-time objective intelligibility measure for time-frequency weighted noisy speech. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 4214–4217.