Interspeech 2024
1-5 September 2024, Kos, Greece

Blind Zero-Shot Audio Restoration: A Variational Autoencoder Approach for Denoising and Inpainting

Veranika Boukun, Jakob Drefs, Jörg Lücke

Machine Learning Lab
Department of Medical Physics and Acoustics, Carl von Ossietzky Universität Oldenburg, Germany
veranika.boukun@uol.de, jakob.drefs@uol.de, joerg.luecke@uol.de

Abstract

We address the task of blind 'zero-shot' audio signal denoising and inpainting. In the blind zero-shot setting, only the corrupted audio signal is used for signal restoration (no other signals are available to train the model). For this challenging setting, we apply a recent variational autoencoder that can leverage advanced probabilistic variational optimization in addition to flexible data modeling enabled by deep neural networks (DNNs). The investigated approach uses a non-amortized encoder and truncated posteriors as variational distributions. This way, the posterior correlations can be approximated, and a theoretically grounded treatment of missing values is directly available. In benchmarks for denoising and inpainting and in comparison with other zero-shot approaches, we observe competitive performance. Our results suggest that combining high-quality probabilistic optimization with DNN optimization is a very promising strategy for challenging audio restoration tasks.
Index Terms: variational autoencoders, zero-shot audio denoising, zero-shot audio inpainting, variational optimization

1. Introduction

Denoising and inpainting (a.k.a. missing data imputation) are two standard audio restoration tasks (e.g., [1, 2]). Conventional neural network approaches often require large amounts of data and (ideally high quality) label information in the form of 'clean' (i.e., non-noisy) data. However, requiring clean data can be a severe limitation for many reasons: (A) it is often expensive and time-consuming to obtain the required data labels [3, 4], (B) clean data is often difficult to obtain or not at all available in real-world scenarios [5], (C) there can potentially be a domain mismatch when synthetic data is used for training [6], (D) artifacts can be introduced [7] and (E) it can be very difficult to obtain a balanced dataset [8].

Approaches applicable in the blind zero-shot setting offer a remedy to the problems mentioned above. The setting (1) means that methods have to be trainable on noisy data; and (2) they have to be trainable exclusively on the the data that one seeks to restore. Methods fulfilling these conditions leverage intrinsic data statistics (e.g., [9] for a discussion) for data restoration. As the first condition also represents a desirable property on its own, methods trainable on noisy data have moved into the focus of ongoing research (e.g., [10, 11]). And full zero-shot methods have by now repeatedly been used, e.g., for audio denoising [1, 12] as well as for visual data denoising [10, 11, 13].

To allow for learning without clean data, different strategies have been suggested. For instance, the objective for conventional DNN learning has been augmented [10, 11]. Alternatively, deep probabilistic generative models such as variational autoencoders (VAEs) [14] can be used because of their ability to learn unsupervised (without clean data). Conventional DNN approaches, their augmented versions for noisy data as well as conventional VAEs usually owe their capabilities to large DNNs with many parameters and hyperparameters. However, large numbers of DNN parameters and hyperparameters usually require large amounts of (sufficiently high quality) data. For the here addressed setting, we therefore apply a novel VAE approach [15] which can provide high performance using relatively small DNNs. Performance of the approach rests, in large parts, on a novel variational optimization [15, 16] which seeks to extract as much information as possible from each data point. The property matches the here investigated setting well, and we consequently explore the method's properties for blind zero-shot audio restoration.

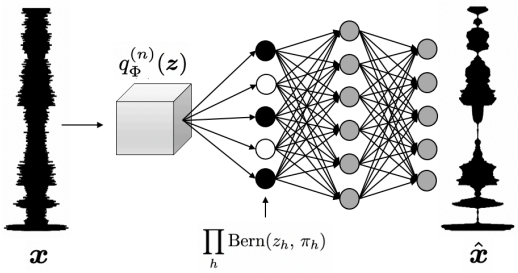

Figure 1: *Schematic representation of audio restoration (here: audio denoising) using TVAE.*

2. Truncated Variational Autoencoder

Given the data $\boldsymbol{x}^{(1:N)}$, we follow a maximum likelihood (ML) approach to estimate optimal parameters of a probabilistic data model $p_\Theta(\boldsymbol{x})$. Concretely, we seek parameters Θ^*, such that $\Theta^* = \operatorname{argmax}_\Theta \mathcal{L}(\Theta) = \operatorname{argmax}_\Theta \sum_n \log p_\Theta(\boldsymbol{x}^{(n)})$. The generative model (i.e., the decoder) we assume to be given by [15]:

$$p_\Theta(\boldsymbol{z}) = \prod_h \left(\pi_h^{z_h}(1-\pi_h)^{(1-z_h)}\right), \tag{1}$$

$$p_\Theta(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}\big(\boldsymbol{x}; \boldsymbol{\mu}(\boldsymbol{z}; W), \sigma^2\mathbb{I}\big), \tag{2}$$

where $\boldsymbol{z} \in \{0,1\}^H$ is a binary latent variable, $\boldsymbol{\pi} \in [0,1]^H$ are the parameters of a Bernoulli prior distribution $p_\Theta(\boldsymbol{z})$ and $\boldsymbol{\mu}(\boldsymbol{z}; W)$ (for conciseness purposes, we will use $\boldsymbol{\mu}(\boldsymbol{z})$ further) is a DNN (W being the weights and biases), which determines the mean of the Gaussian conditional distribution. For the observable distribution $p_\Theta(\boldsymbol{x}|\boldsymbol{z})$, we assume homoscedasticity of the Gaussian distribution. The set of all model parameters is then given by $\Theta = \{\boldsymbol{\pi}, W, \sigma^2\}$.

10.21437/Interspeech.2024-314

Instead of optimizing the log-likelihood directly, VAEs optimize a variational lower bound (ELBO) [17]:

$$\mathcal{F}(\Phi,\Theta) = \sum_n \mathbb{E}_{q_\Phi^{(n)}}\big[\log\big(p_\Theta(\boldsymbol{x}^{(n)}|\boldsymbol{z})\big)\big] - \sum_n \mathrm{D_{KL}}\big[q_\Phi^{(n)}(\boldsymbol{z}); p_\Theta(\boldsymbol{z})\big], \quad (3)$$

where $q_\Phi^{(n)}(\boldsymbol{z})$ are the variational distributions with parameters $\Phi = (\Phi^{(1)}, ..., \Phi^{(N)})$, $\mathbb{E}_{q_\Phi^{(n)}}$ is the expectation w.r.t. $q_\Phi^{(n)}$ and $\mathrm{D_{KL}}$ is the Kullbach-Leibler divergence.

Optimization of the Encoding Model. VAEs with discrete latents usually require specific treatments to allow for gradient-based learning, such as REINFORCE [18] with variance control techniques [19] or 'Gumbel-softmax' approximation [20]. As in previous work [15], we use a direct evolutionary optimization of the encoding model $q_\Phi^{(n)}$. In this procedure, truncated posteriors [21] are chosen as an alternative to a DNN-based encoder, which avoids gradient propagation through discrete latents and an amortization gap [22]. As additional advantage for audio inpaining, such an encoder can treat missing values directly as unknown observables [15], whereas in amortized approaches, treatment of missing values has to be addressed by defining masks or zero-padding (which can be problematic).

Truncated posterior approximations assume that the posterior mass can be represented well by a small subset of the latent space. The family of truncated variational distributions is defined as follows [16, 21], where δ is an indicator function:

$$q_\Phi^{(n)}(\boldsymbol{z}) \coloneqq \frac{p_\Theta(\boldsymbol{z}|\boldsymbol{x}^{(n)})}{\sum_{\boldsymbol{z'}\in\Phi^{(n)}} p_\Theta(\boldsymbol{z'}|\boldsymbol{x}^{(n)})}\,\delta(\boldsymbol{z}\in\Phi^{(n)}) = \frac{p_\Theta(\boldsymbol{x}^{(n)},\,\boldsymbol{z})}{\sum_{\boldsymbol{z'}\in\Phi^{(n)}} p_\Theta(\boldsymbol{x}^{(n)},\,\boldsymbol{z'})}\,\delta(\boldsymbol{z}\in\Phi^{(n)}), \quad (4)$$

with $\Phi^{(n)}$ being the subset of latent states $\boldsymbol{z}$, i.e., for all $\boldsymbol{z}\in\Phi^{(n)}$ the probability $q_\Phi^{(n)}(\boldsymbol{z})$ is proportional to $p_\Theta(\boldsymbol{z}|\boldsymbol{x}^{(n)})$ and zero for all $\boldsymbol{z}\notin\Phi^{(n)}$. $p_\Theta(\boldsymbol{x}^{(n)},\,\boldsymbol{z})$ is the joint distribution, $p_\Theta(\boldsymbol{x}^{(n)},\,\boldsymbol{z}) = p_\Theta(\boldsymbol{x}^{(n)}|\boldsymbol{z})p_\Theta(\boldsymbol{z})$ (joint notation will be used from here on). The use of truncated posteriors allows for the reformulation of the variational lower bound in Eq. (3), i.e., the bound becomes equal to (compare [16]):

$$\mathcal{F}(\Phi,\Theta) = \sum_n \log\Big(\sum_{\boldsymbol{z}\in\Phi^{(n)}} p_\Theta(\boldsymbol{x}^{(n)},\,\boldsymbol{z})\Big). \quad (5)$$

The reformulation in Eq. (5) offers an efficient way of optimizing the bound w.r.t. $\Phi^{(n)}$. Concretely, the functional form of Eq. (5) means that $\mathcal{F}(\Phi,\Theta)$ is increased if we replace a state $\boldsymbol{z}^{\mathrm{old}}\in\Phi^{(n)}$ by a new state $\boldsymbol{z}^{\mathrm{new}}\notin\Phi^{(n)}$, such that:

$$p_\Theta(\boldsymbol{x}^{(n)},\boldsymbol{z}^{\mathrm{new}}) > p_\Theta(\boldsymbol{x}^{(n)},\boldsymbol{z}^{\mathrm{old}}). \quad (6)$$

As in [23], we combine variational optimization with evolutionary optimization (EVO): first, candidates for new states are generated by mutation and crossover based on the population of old states $\boldsymbol{z}^{\mathrm{old}}$. Then new states $\boldsymbol{z}^{\mathrm{new}}$ are selected using the criterion of Eq. (6) (see [15, 23] for details).

Optimization of the Decoding Model. Using the truncated variational distributions $q_\Phi^{(n)}(\boldsymbol{z})$, the gradient of the objective in Eq. (5) with respect to W can be computed as [15]:

$$\boldsymbol{\nabla}_W\mathcal{F}(\Phi,\Theta) = -\frac{1}{2\sigma^2}\sum_{n,\boldsymbol{z}\in\Phi^{(n)}} q_\Phi^{(n)}(\boldsymbol{z})\boldsymbol{\nabla}_W||\boldsymbol{x}^{(n)}-\boldsymbol{\mu}(\boldsymbol{z})||^2.$$

As a consequence, standard automatic differentiation tools can be used for the decoder optimization. The update equations for the remaining model parameters σ^2 and $\boldsymbol{\pi}$ are given by [15]:

$$\sigma^2 = \frac{1}{D\,N}\sum_n\sum_{\boldsymbol{z}\in\Phi^{(n)}} q_\Phi^{(n)}(\boldsymbol{z})||\boldsymbol{x}^{(n)}-\boldsymbol{\mu}(\boldsymbol{z})||^2,$$

$$\boldsymbol{\pi} = \frac{1}{N}\sum_n\sum_{\boldsymbol{z}\in\Phi^{(n)}} q_\Phi^{(n)}(\boldsymbol{z})\,\boldsymbol{z}. \quad (7)$$

To the model and the training procedure described in this section we will refer to as Truncated Variational Autoencoder (TVAE) [15]. A schematic representation of the TVAE can be seen in Fig. 1.

Audio Restoration. We use corrupted audio signals of 1-2 s sampled at 16 kHz. The corrupted signals are restored using chunk-based estimation: First, we extract all chunks of length T (here T is 25 ms), while allowing for overlapping chunks (we use maximal possible overlap, i.e., a stride of one). Signal length, chunk length T and stride determine the number N of different extracted chunks which are used as training set. Encoder and decoder parameters Φ and Θ are then inferred from the training set using the procedure described above (in the following Φ and Θ will refer to the optimized parameters). Given the tth discrete time sample x_t (amplitude value) of the corrupted signal $\boldsymbol{x}$, we can use a chunk $\boldsymbol{x}^{(n)}$ which contains x_t to estimate the amplitude of the underlying clean signal [23]:

$$x_t^{\mathrm{est}} = \mathbb{E}_{p_\Theta(\boldsymbol{z}|\boldsymbol{x})}[\mu_t(\boldsymbol{z})]. \quad (8)$$

The exact posterior $p_\Theta(\boldsymbol{z}|\boldsymbol{x})$ required for this estimate is not computationally tractable. However, we can use the encoder $q_\Phi^{(n)}$ of the VAE as approximation and obtain:

$$x_t^{\mathrm{est}} \approx \mathbb{E}_{q_\Phi^{(n)}}[\mu_t(\boldsymbol{z})] = \frac{\sum_{\boldsymbol{z}\in\Phi^{(n)}} p_\Theta(\boldsymbol{x}^{(n)},\boldsymbol{z})\,\mu_t(\boldsymbol{z})}{\sum_{\boldsymbol{z'}\in\Phi^{(n)}} p_\Theta(\boldsymbol{x}^{(n)},\boldsymbol{z'})}. \quad (9)$$

A time sample t is contained in several chunks (for stride one in T chunks except for the signal start and end). To obtain the final estimate $\hat{x}_t$, we average over all estimates x_t^{est} that are provided by the chunks containing t. This procedure is then executed for all t time samples to obtain the restored signal $\hat{\boldsymbol{x}}$.

For inpainting we proceed analogously. Only the posterior in Eq. (8) and its approximation in Eq. (9) are computed w.r.t. the non-missing data of each chunk (see [23] for details).

3. Experiments

We benchmark TVAE-based restoration in the blind zero-shot setting for audio denoising and audio inpainting.

Baselines. For audio denoising with additive Gaussian noise, two baselines were selected, namely, a Geometric Approach (GA) [24] and Minimum-Mean Square Error Short-Time Spectral Amplitude Estimator (MMSE-STSA) [25] with the noise power spectral density estimator from [26]. Additionally, a U-Net-based architecture with dilated convolutions, namely the 'Deep Audio Prior with Dilated Convolutions' (we will further refer to as DCDAP) [1] was chosen for comparison. In audio inpainting, we use a benchmark recently suggested by Turetzky et al. [2] (see their Sec. 5 and Tab. 2). The benchmark allows for a comparison with two 'Deep Audio Waveform Prior' approaches (we will further refer to as DAWP), which use a modified Demucs architecture[1] and a WaveUNet [27].

[1] A. Defossez et al., "Music Source Separation in the Waveform Domain," arXiv preprint, arXiv:1911.13254, 2019.

Datasets. Following the experimental design in [1], LJ-Speech[2], SC09 Spoken Numbers[3] [28, 29] and Bach Performances [28] datasets were used in audio denoising experiments. LJ-Speech is an open-source dataset including 13300 audio clips (1-10 seconds at 22 kHz) of read passages from 7 non-fictional books. SC09 Spoken Numbers dataset includes audio clips ($\sim$1 second at 16 kHz) of digits 0-9 spoken by a range of speakers with various accents. Bach performances dataset includes piano recordings of Bach compositions (audio clip duration >50 seconds at 48 kHz), 0.3 hours in total. Following the experimental design in [2], speech from VoiceBank [30], singing from MUSDB18[4] and music from MedleyDB 2.0 [31] datasets were used in the audio inpainting experiments. VoiceBank includes clean and noisy utterances of English speakers at 48 kHz. MUSDB18 consists of 150 songs (individual instruments, vocals and mix tracks as stereo signals at 44.1 kHz). MedleyDB 2.0 includes 74 multi-track songs of different genre provided at 44.1 kHz.

3.1. Evaluation Metrics

PSNR and PESQ were used as evaluation metrics in audio denoising experiments as proposed in [1]. The PSNR is defined as follows:

$$\mathrm{PSNR}(\boldsymbol{s},\, \hat{\boldsymbol{x}}) = 10\, \log_{10} \left(\frac{\mathrm{MAX}_I^2}{\mathrm{MSE}(\boldsymbol{s},\, \hat{\boldsymbol{x}})} \right), \qquad (10)$$

where MAX_I denotes the difference between the maximum and the minimum amplitudes of the signals $\in\ (-1, 1)$ and $\mathrm{MSE}(\boldsymbol{s},\, \hat{\boldsymbol{x}})$ is the mean-squared error between the clean signal $\boldsymbol{s}$ and $\hat{\boldsymbol{x}}$. The scikit-image implementation was used to calculate PSNR. In order to measure the PESQ scores, python-pesq library was used (option 'wb').

In addition to PSNR, SI-SNR [2] was considered as evaluation metric. To compute SI-SNR, we used the torchmetrics implementation, where SI-SNR is defined as in [32]:

$$\text{SI-SNR}(\boldsymbol{s},\, \hat{\boldsymbol{x}}) = 10\, \log_{10} \frac{||\tilde{\boldsymbol{x}}||^2}{||\boldsymbol{e}||^2}, \qquad (11)$$

where $\tilde{\boldsymbol{x}} = \dfrac{\langle \hat{\boldsymbol{x}},\, \boldsymbol{s} \rangle\, \boldsymbol{s}}{||\boldsymbol{s}||^2}$ and $\boldsymbol{e} = \hat{\boldsymbol{x}} - \boldsymbol{s}$.

3.2. Implementation Details

In all experiments, overlapping chunks of $T = 400$ time samples were extracted from corrupted audio signals, corresponding to $N = 15601$ chunks (for 1 s signal) and $N = 31601$ chunks (for 2 s). The number of variational parameters $S = |\Phi^{(n)}| = 64$, hence, $N \times (|\Phi^{(n)}| + |\Phi^{(n)}_{\mathrm{new}}|)$ states are evaluated per epoch. Evolutionary optimization (EVO) used 5 parents, 9 children and 4 generations, 'uniform' mutation, no crossover and the parent selection based on 'fitness' (Eq. 6). We used a decoder DNN with three fully-connected layers and ReLU activations. We used $H = 64$ neurons for the first layer (which corresponds to the number of prior variables), 512 neurons for the middle layer, and 400 neurons for the output layer (which is determined by the chunk size T). The prior parameters were initialized as $\pi_h = 1/H$, σ^2 was initialized as mean of the variances of the data points plus a constant factor 10^{-3}. The DNN weights were initialized as samples of Xavier normal distribution (gain factor of 1), and the biases as 0. A cycling learning rate [33] and Adam optimizer were used for decoder optimization, with the lowest and the highest learning rates of 10^{-4} and 10^{-3}, correspondingly, and 20 iterations in the increasing half of a cycle. The batch size $B = 32$ was used and the number of epochs was 500. In all experiments, the audio was resampled at 16 kHz to allow for a fair comparison with [1] and [2]. The used GPU hardware included: NVIDIA Geforce GTX Titan X and Titan X Pascal, NVIDIA Tesla P100 and V100 GPUs.

Table 1: *Comparison of blind zero-shot audio denoising performance (additive Gaussian noise). Δ denotes the score improvement with respect to the noisy audio signal. $\varnothing$ denotes the average over 3 runs. As a statistical measure, we report population standard deviation together with the obtained results.*

	LJ-Speech		**SC09**		**Bach**
	ΔPSNR	ΔPESQ	ΔPSNR	ΔPESQ	ΔPSNR
GA	6.17 $\pm$1.30	0.02 $\pm$0.01	8.14 $\pm$1.02	0.05 $\pm$0.03	10.50 $\pm$1.08
MMSE-STSA	**9.04** $\pm$1.99	**0.07** $\pm$0.02	8.85 $\pm$5.92	**0.18** $\pm$0.15	**13.30** $\pm$2.08
DCDAP	8.51 $\pm$0.88	0.03 $\pm$0.02	**11.02** $\pm$1.49	0.03 $\pm$0.01	10.40 $\pm$1.78
TVAE$^{\varnothing}$	**12.26** $\pm$0.95	**0.48** $\pm$0.19	**14.99** $\pm$1.17	**0.25** $\pm$0.21	**16.06** $\pm$1.34

3.3. Audio Denoising

In this task, we applied TVAE to a noisy audio signal, which was produced by adding Gaussian noise with a standard deviation $\sigma = 0.1$ to the clean audio signal. To reduce computation times, we randomly selected five audio clips from each used dataset, i.e. from LJ-Speech, SC09, and Bach Performances. These (in total 15) audio clips we used for the evaluation of all methods. The results are presented in Tab. 1. GA, MMSE-STSA and DCDAP are each applied once to each audio clip which results in five PSNR and PESQ scores per dataset. Tab. 1 reports the corresponding average PSNR and PESQ improvements for each dataset. The TVAE method was applied three times to each audio clip resulting in 15 PSNR and PESQ scores per dataset. In Tab. 1 the average PSNR and PESQ improvements for each dataset are reported. For non-speech Bach Performances, we report only the PSNR improvements.

For all three datasets, TVAE can improve the performance of the baseline methods in terms of both PSNR and PESQ (see Fig. 2 for visual result comparison). We did not compare to the recent Only-Noisy Training (ONT) method [34] because the data and settings for our benchmark in Tab. 1 where unsuitable for that approach: (A) ONT is optimized for speech, while the methods of Tab. 1 are applicable to general audio signal (including music); and (B) the sampling rate of 16 kHz of the benchmark in Tab. 1 did not match the sampling rate of 48 kHz which ONT uses (we used the implementation[5] for our investigations). Due to the different expected data and settings, we found that amendments of the code would be necessary that would significantly go beyond changing preprocessing and hyperparameter settings.

3.4. Audio Inpainting

To investigate inpainting performance, we adapted the experimental procedure from [2] in order to compare the performance of TVAE to DAWP training with Demucs and WaveUNet architectures. The corruption process included masking a small, non-

[2]https://keithito.com/LJ-Speech-Dataset/

[3]P. Warden, "Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition," arXiv preprint, arXiv:1804.03209, 2018.

[4]Z. Rafii et al., "The MUSDB18 corpus for music separation," Dec. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.1117372

[5]https://github.com/liqingchunnnn/Only-Noisy-Training

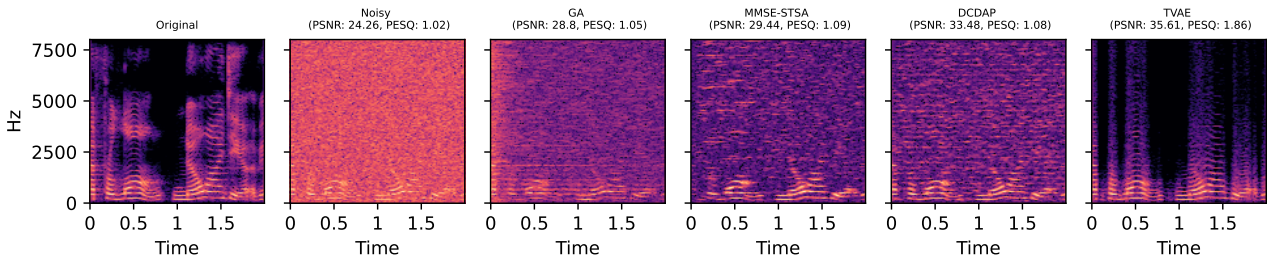

Figure 2: *Results of TVAE audio signal denoising compared to other baselines. The audio clip was taken from the LJ-Speech dataset. Spectrograms are only used for illustration purposes.*

silent segment in an audio clip. The missing segment lengths include 1, 2 and 5 ms (corresponding to 16, 32 and 80 consecutive time samples missing at 16 kHz).

The preprocessing included cutting the available audio signals into non-silent audio clips of 2 s, following [2]. The speech data was taken from the 28-speaker VoiceBank clean trainset. For singing data, 5 songs were randomly chosen from the test set of MUSDB18 dataset and the corresponding 5 vocal tracks were preprocessed. For music data, 4 compositions were randomly chosen from Medley 2.0 and only the corresponding instrumental tracks (3 included mix of instruments and one single instrument track) were preprocessed. From the preprocessed files, 20 were randomly chosen from each dataset. To reduce computational costs, the TVAE method was applied once to each audio clip in every condition (1, 2 and 5 ms missing) resulting in 60 SI-SNR and PSNR scores per dataset. In Tab. 2, the average SI-SNR and PSNR scores for each dataset are reported. For DAWP approaches, SI-SNR and PSNR scores reported in Tab. 2 are taken directly from [2].

Table 2: *Zero-shot audio inpainting performance for TVAE and the DAWP baselines. As a statistical measure, we report population standard deviation together with the obtained results.*

		SI-SNR			PSNR		
		WUnet	Demucs	TVAE	WUnet	Demucs	TVAE
1 ms	Speech	-2.05	6.57	**13.75** ±2.16	12.02	32.71	**42.79** ±1.31
	Music	-1.83	6.81	**12.58** ±8.84	11.97	34.46	**43.52** ±4.90
	Singing	-3.73	4.21	**10.90** ±2.77	11.95	**38.40**	36.32 ±4.27
2 ms	Speech	-5.33	3.55	**13.69** ±2.21	10.53	34.02	**42.72** ±1.29
	Music	-5.77	2.98	**12.69** ±8.26	10.79	30.44	**43.48** ±4.87
	Singing	-8.45	-2.35	**10.91** ±2.78	10.02	31.62	**36.32** ±4.28
5 ms	Speech	-12.13	-2.60	**13.76** ±2.16	9.73	30.46	**42.81** ±1.35
	Music	-9.48	-2.11	**12.70** ±8.02	10.47	30.03	**43.46** ±4.93
	Singing	-11.07	-4.60	**10.92** ±2.75	10.15	31.46	**36.34** ±4.21

As can be observed (Tab. 2), TVAE shows an improved performance over DAWP approaches in most conditions (excluding "Singing" with 1 ms masked segment). The high population standard deviation of TVAE scores for "Music" conditions can be attributed to the convergence to a very suboptimal local optima for one of the 20 audio clips. Changing the initialization of the decoder weights (e.g., Xavier normal distribution gain parameter set to 0.5) resulted in better convergence and consequently improved results. However, for better comparison, we decided to keep the same weight initialization (gain parameter set to 1) for all experiments. We did not compare to the SPAIN method [35, 36] as it is exclusively focusing on inpainting, and as the original approach [35] was not data-driven. A visual illustration of the audio inpainting achieved by TVAE can be seen in Fig. 3.

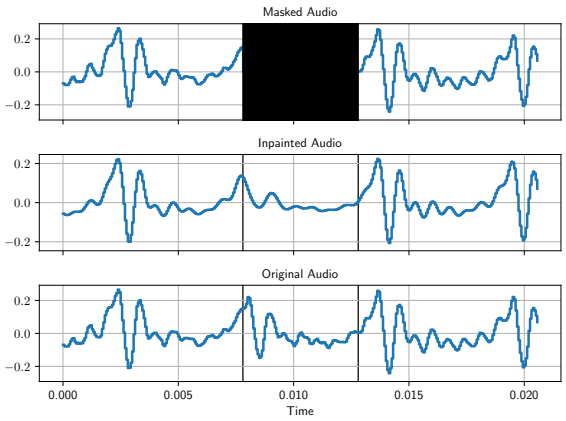

Figure 3: *Resulting speech signal inpainting using TVAE. Zero-masked audio signal (80 consecutive time samples missing), the inpainted and the original audio signals are compared.*

4. Conclusion

We have investigated a recent VAE approach for audio restoration. Evaluation on benchmarks showed that TVAE can significantly improve over competing approaches for both audio denoising (Tab. 1) and audio inpainting (Tab. 2). The observed performance suggests that combining flexible and high-quality probabilistic approximations with DNN-based modeling can partly significantly improve restored audio. TVAE acquires its performance based on a relatively small DNN (architecture 64-512-400) for all experiments, parameter efficiency of TVAE with ∼250 k parameters is thus significantly higher than, e.g., for DCDAP, Demucs and WaveUNet networks with ∼1.1, ∼1.7 and ∼70 million parameters, respectively.

However, performance and parameter efficiency of TVAE also come at a cost. To restore a 2 s signal, TVAE requires ∼7.3 hours, which is long compared to approaches leveraging larger DNNs (and which confirms long execution times reported previously [15]). For TVAE, the high computational demands are not only due to using many chunks and the averaging used for restoration, but also due to combinatorial, non-amortized learning procedure. TVAE consequently offers itself for offline corrupted signal restoration, while other approaches are preferable if fast or compute efficient approaches are required.

5. Acknowledgements

We acknowledge support by the German Research Foundation (DFG) under grant ID 352015383 (SFB 1330, B2). Furthermore, we would like to thank Marvin Tammen for discussions and for providing the source code for MMSE-STSA and GA models as well as Hamid Mousavi for many fruitful discussions and help with this paper.

6. References

[1] V. S. Narayanaswamy, J. J. Thiagarajan, and A. Spanias, "On the Design of Deep Priors for Unsupervised Audio Restoration," *Interspeech*, 2021b.

[2] A. Turetzky *et al.*, "Deep Audio Waveform Prior," *Interspeech*, pp. 2938–2942, 2022.

[3] P. Ma *et al.*, "Auto-AVSR: Audio-Visual Speech Recognition with Automatic Labels," in *IEEE ICASSP*, 2023, pp. 1–5.

[4] J.-H. Choi *et al.*, "Extending Self-Distilled Self-Supervised Learning For Semi-Supervised Speaker Verification," in *IEEE ASRU*, 2023, pp. 1–8.

[5] L.-W. Chen *et al.*, "A Training and Inference Strategy Using Noisy and Enhanced Speech as Target for Speech Enhancement without Clean Speech," in *Interspeech*, 2023, pp. 5315–5319.

[6] X. Liu *et al.*, "SynthVSR: Scaling Up Visual Speech Recognition With Synthetic Supervision," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 18 806–18 815.

[7] T.-Y. Hu *et al.*, "Synt++: Utilizing Imperfect Synthetic Data to Improve Speech Recognition," in *IEEE ICASSP*, 2022, pp. 7682–7686.

[8] S. A. Sheikh *et al.*, "Advancing Stuttering Detection via Data Augmentation, Class-Balanced Loss and Multi-Contextual Deep Learning," *IEEE Journal of Biomedical and Health Informatics*, 2023.

[9] A. Shocher, N. Cohen, and M. Irani, "Zero-Shot" Super-Resolution Using Deep Internal Learning," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 3118–3126.

[10] J. Lehtinen *et al.*, "Noise2Noise: Learning Image Restoration without Clean Data," in *ICML*. PMLR, 2018, pp. 2965–2974.

[11] A. Krull, T.-O. Buchholz, and F. Jug, "Noise2Void-Learning Denoising from Single Noisy Images," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 2129–2137.

[12] Z. Zhang *et al.*, "Deep Audio Priors Emerge From Harmonic Convolutional Networks," in *ICLR*, 2020.

[13] J. Lequyer *et al.*, "A Fast Blind Zero-Shot Denoiser," *Nature Machine Intelligence*, vol. 4, no. 11, pp. 953–963, 2022.

[14] M. Prakash, A. Krull, and F. Jug, "Fully Unsupervised Diversity Denoising with Convolutional Variational Autoencoders," in *ICLR*, 2021.

[15] J. Drefs, E. Guiraud, F. Panagiotou, and J. Lücke, "Direct Evolutionary Optimization of Variational Autoencoders with Binary Latents," in *ECML PKDD*. Springer, 2023, pp. 357–372.

[16] F. Hirschberger, D. Forster, and J. Lücke, "A variational em acceleration for efficient clustering at very large scales," *IEEE TPAMI*, vol. 44, no. 12, pp. 9787–9801, 2022.

[17] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *ICLR*, 2014.

[18] R. J. Williams, "Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning," *Machine Learning*, vol. 8, pp. 229–256, 1992.

[19] A. Dimitriev and M. Zhou, "CARMS: Categorical-Antithetic-Reinforce Multi-Sample Gradient Estimator," *NeurIPS*, vol. 34, pp. 13 217–13 229, 2021.

[20] E. Jang, S. Gu, and B. Poole, "Categorical Reparameterization with Gumbel-Softmax," in *ICLR*, 2017.

[21] J. Lücke and J. Eggert, "Expectation Truncation and the Benefits of Preselection in Training Generative Models," *JMLR*, vol. 11, pp. 2855–2900, 2010.

[22] C. Cremer, X. Li, and D. Duvenaud, "Inference Suboptimality in Variational Autoencoders," in *ICML*. PMLR, 2018, pp. 1078–1086.

[23] J. Drefs, E. Guiraud, and J. Lücke, "Evolutionary Variational Optimization of Generative Models," *JMLR*, vol. 23, no. 1, pp. 935–985, 2022.

[24] Y. Lu and P. C. Loizou, "A Geometric Approach To Spectral Subtraction," *Speech Communication*, vol. 50, no. 6, pp. 453–466, 2008.

[25] Y. Ephraim and D. Malah, "Speech Enhancement using a Minimum-Mean Square Error Short-Time Spectral Amplitude Estimator," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 32, no. 6, pp. 1109–1121, 1984.

[26] T. Gerkmann and R. C. Hendriks, "Unbiased MMSE-based Noise Power Estimation with Low Complexity and Low Tracking Delay," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 20, no. 4, pp. 1383–1393, 2012.

[27] D. Stoller, S. Ewert, and S. Dixon, "Wave-U-Net: A Multi-Scale Neural Network for end-to-end Audio Source Separation," *ISMIR*, p. 334–340, 2018.

[28] C. Donahue, J. McAuley, and M. Puckette, "Adversarial Audio Synthesis," *ICLR*, 2018.

[29] K. Goel *et al.*, "It's Raw! Audio Generation with State-Space Models," *ICML*, 2022.

[30] C. Valentini-Botinhao *et al.*, "Noisy Speech Database for Training Speech Enhancement Algorithms and TTS Models, 2016 [sound]," *University of Edinburgh. School of Informatics, CSTR*, 2017.

[31] R. M. Bittner *et al.*, "MedleyDB 2.0: New Data and a System for Sustainable Data Collection," *ISMIR*, p. 36, 2016.

[32] Y. Luo and N. Mesgarani, "TaSNet: Time-Domain Audio Separation Network for Real-Time, Single-Channel Speech Separation," in *IEEE ICASSP*, 2018, pp. 696–700.

[33] L. N. Smith, "Cyclical Learning Rates for Training Neural Networks," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017, pp. 464–472.

[34] J. Wu *et al.*, "Self-supervised speech denoising using only noisy audio signals," *Speech Communication*, vol. 149, pp. 63–73, 2023.

[35] O. Mokrỳ *et al.*, "Introducing Spain (SParse Audio Inpainter)," in *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE, 2019, pp. 1–5.

[36] G. Tauböck, S. Rajbamshi, and P. Balazs, "Sparse Audio Inpainting: A Dictionary Learning Technique to Improve Its Performance," in *Audio Engineering Society Convention 149*. Audio Engineering Society, 2020.