Drone Ego-Noise Cancellation for Improved Speech Capture using Deep Convolutional Autoencoder Assisted Multistage Beamforming

Yanjue Song, Stijn Kindt, Nilesh Madhu
IDLab, Ghent University - imec
Ghent, Belgium
yanjue.song@ugent.be, stijn.kindt@ugent.be, nilesh.madhu@ugent.be

***Abstract*—We propose a multistage approach for enhancing speech captured by a drone-mounted microphone array. The key challenge is suppressing the drone ego-noise, which is the major source of interference in such captures. Since the location of the target is not known *a priori*, we first apply a UNet-based deep convolutional autoencoder (AE) individually to each microphone signal. The AE generates a time-frequency mask $\in [0, 1]$ per signal, where high values correspond to time-frequency points with relatively good signal-to-noise ratios (SNRs). The masks are pooled across all microphones and the aggregated mask is used to steer an adaptive, frequency domain beamformer, yielding a signal with an improved SNR. This beamformer output, after being fed back to the AE, now yields an improved mask – which is used for re-focussing the beamformer. This combination of AE and beamformer, which can be applied to the signals in multiple 'passes' is termed multistage beamforming. The approach is developed and evaluated on a self-collected database. For the AE - when used to steer a beamformer - a training target that preserves more speech at the cost of less noise suppression outperforms an aggressive training target that suppresses more noise at the cost of more speech distortion. This, in combination with max-pooling of the multi-channel mask – which also lets through more speech (and noise) compared with median pooling – performs best. The experiments further demonstrate that the multistage approach brings extra benefit to the speech quality and intelligibility when the input SNR is $\geq -10\,$dB, and yields comprehensible outputs when the input has a SNR above $-5\,$dB.**

***Index Terms*—Drone, unmanned aerial vehicle (UAV), ego-noise reduction, beamformer, speech enhancement, UNet, MVDR, MWF, autoencoders.**

I. Introduction

Unmanned aerial vehicles (drones) are becoming increasingly affordable and accessible. Video capturing is one of their most natural and prominent applications. Capturing the audio can also be helpful, e.g., in search and rescue tasks; however, the strong ego-noise from the drone engines and the wind disturbance lead to extremely low signal-to-noise ratios (SNRs) in the captured signals, limiting the usability of such captures. Drone noise is also highly non-stationary because the rotation speed of the engines changes frequently during flight and even during hover. Together, all these factors present a formidable challenge for drone-based audio scene capturing.

Due to low prevailing SNRs, multi-microphone methods [1]–[6] are more common than single channel approaches [7], [8] in drone ego-noise removal. In fact, auxiliary information such as propeller rotation speed [8] or drone noise reference signals [7] is usually required by these single-microphone methods. Even so, they perform poorly for real recordings due to the non-stationary noise conditions [7]. In contrast, multi-microphone approaches can exploit spatial information in these situations, leading to targeted noise suppression by beamforming [1], [2], [6] or by blind source separation [5].

For a drone, the predominant source of interference arises from the spatially-localised ego-noise from the (typically) nearby rotors. Approaches such as the adaptive MVDR [9] take the estimated noise spectrum into account when computing the beamformer weights, which improves cancellation of ego-noise. [1] uses a minimum variance distortionless response (MVDR) beamformer steered towards the target location. The noise covariance matrices were *fixed* and designed to cancel signals coming from the rotor directions. Additional beamformers, steered toward the rotors, were used to estimate a single channel Wiener postfilter. Multi-channel Wiener filters (MWF) [10], [11] implicitly combine a postfilter with an adaptive MVDR beamformer, leading to improved residual-noise suppression.

Such adaptive beamformers and postfilters are usually implemented in the short-time frequency domain, where they are formulated in terms of the spatial covariance matrices (SCMs) of the respective signals [12]. The target-speech SCM and noise SCM are not known *a priori* but are estimated from the noisy input signals. The estimation is done independently at each frequency, and typically obtained by a weighted recursive average. The weights (or masks) for each time-frequency (TF) point are real-valued and $\in [0, 1]$. They indicate how dominant speech or the noise is at a time-frequency point. For good spatial filtering, the key lies, thus, in the reliable estimation of these masks. For example, in [2], the masks are based on the direction of arrival (DoA) estimates in each TF bin. Using the recent advances in deep learning, the same authors use a feed forward neural network to further enhance the DoA masks [6]. Deep neural networks can also be exploited to directly estimate the masks [6], [13], [14].

Deep convolutional autoencoders (AE) with skip connec-

This work is supported by the Research Foundation - Flanders (FWO) under grant number G081420N.

tions, also known as UNets due to their topology, have recently established themselves in speech enhancement [15]. Such networks learn to estimate either the underlying clean signal spectrogram or the so-called ideal ratio mask (IRM) from the noisy input spectrogram. UNets can be trained for single- or multi-microphone recordings.

Here, we propose to enhance the audio captured by a drone-mounted microphone array by a multistage adaptive beamformer, which is steered by time-frequency masks obtained from a single-channel AE. The choice of a single-channel AE to obtain the time-frequency mask, despite the availability of multi-channel signals, is a conscious one for the following two reasons: firstly, training multi-channel UNets requires implicit or explicit incorporation of spatial information. This necessitates the generation of a more exhaustive and diverse training set (either by varying the location of the target source so the UNet does not overfit to a particular direction, or by training the array for an a priori fixed direction). Since the location of the target source is not known *a priori*, a large training dataset is necessary, which is difficult to acquire for UAV-based captures. Secondly, multi-channel UNets, once trained, are specific to the array configuration. Thus, if the array configuration changes (e.g., microphones are damaged or a different array geometry is used), a complete retraining is required, making these approaches less flexible. In contrast, applying a single-channel AE to individual microphone signals is, thus, independent of the array configuration and requires a smaller training dataset.

The multi-channel TF mask by applying the AE to each microphone signal is then pooled to aggregate a single-channel mask to steer the beamformer. The improved signal from the beamformer is subsequently fed back to the AE, yielding a further refined mask. We can thus iterate over the combination of AE and beamformer where, at each stage, the AE produces a refined mask from the previous beamformer output, and the subsequent beamformer stage uses this mask to re-focus and better attenuate the noise and enhance the speech. This iterative process leads to more weak speech components being recovered as compared to a single-stage system. Further we analyse the effect of different training targets in the AE and show how it influences the processing chain. These results may also be applicable in the general field of speech enhancement.

The paper is structured as follows: Section II introduces the signal model and details the proposed method and its components. The approach is evaluated in Section III, where we first describe our data-collection and subsequently provide the evaluation results and discussions. We conclude with some general remarks on this topic, setting up our future work.

II. Proposed Method

A. Signal Model

The signal captured by an M-element microphone array is a mixture of the target speech and interference. Given that drones mostly operate in the open, we may model the mixture at any microphone m as:

$$x_m(n) = h_m(n) * s(n) + v_m(n)\,, \tag{1}$$

where $s(n)$ is the speech signal, $h_m(n)$ is the impulse response modeling the propagation delay and attenuation from the source location to microphone m, $*$ is the convolution operator and $v_m(n)$ is the noise at microphone m. The noise in this case mainly consists of spatially-localised ego-noise generated by the rotors. Using the short term Fourier transform (STFT) representation, we can write (1) as:

$$X_m(l,k) = H_m(k)S(l,k) + V_m(l,k)\,, \tag{2}$$

with l being the frame index and $k = \{0, 1, \ldots, K\}$ being the $K+1$ discrete frequency bin indices from DC to Nyquist (i.e., positive frequency spectrum). The above model assumes that the location of the target speech remains fixed (or varies very slowly) with respect to the array, allowing us to drop the time-dependency on the spatial transfer function H_m. This is not a limiting assumption however – but serves to simplify the exposition. A more compact expression is obtained by stacking the signals from the different microphones in column vectors:

$$\mathbf{X}(l,k) = \mathbf{H}(k)S(l,k) + \mathbf{V}(l,k) \tag{3}$$

Where $\mathbf{H}(k) = [H_1(k), \cdots, H_M(k)]^T$ is the so-called *steering* vector and $\mathbf{V}(l,k) = [V_1(l,k), \cdots, V_M(l,k)]^T$ is the vector of interfering sources at the microphones.

B. Single-channel deep convolutional autoencoder

In single-microphone enhancement, the enhancement function is obtained as a TF mask $\mathcal{M}(l,k) \in [0,1]$, where the magnitude of $\mathcal{M}(l,k)$ is indicative of the degree of speech presence. If directly applied to the STFT spectrum of the noisy signal, the mask suppresses the unwanted noise while keeping the desired speech. The enhanced signal at each channel can then be written as:

$$Y(l,k) = \mathcal{M}_m(l,k)X_m(l,k) \tag{4}$$

These masks, obtained individually for each microphone, can be pooled over microphones to yield a more robust estimate, i.e., the maximum or the median of all masks at each TF bin is aggregated for a new mask:

$$\mathcal{M}(l,k) = \mathrm{Pool}_{m=1}^{M}\big(\mathcal{M}_m(l,k)\big) \tag{5}$$

The mask is estimated by a deep denoising autoencoder, implemented as a UNet architecture (Fig. 1). This architecture, inspired by [15], consists of 11 layers with skip connections. The input to the AE is the sequence of L consecutive frames of the log-magnitude STFT spectrum of the signal, where the last frame in the sequence is the current frame. The training target is the ideal ratio mask (IRM) [16]:

$$\mathcal{M}_m(l,k) \triangleq \mathrm{IRM}_m(l,k) = \sqrt{\frac{|S_m(l,k)|^2}{|S_m(l,k)|^2 + |V_m(l,k)|^2}}\,, \tag{6}$$

where $S_m(l,k) = H_m(k)S(l,k)$ is the target speech at microphone m. We introduce an extra fully-connected (FC) layer at the output, which combines the penultimate layer output to yield the IRM corresponding to the current input frame.

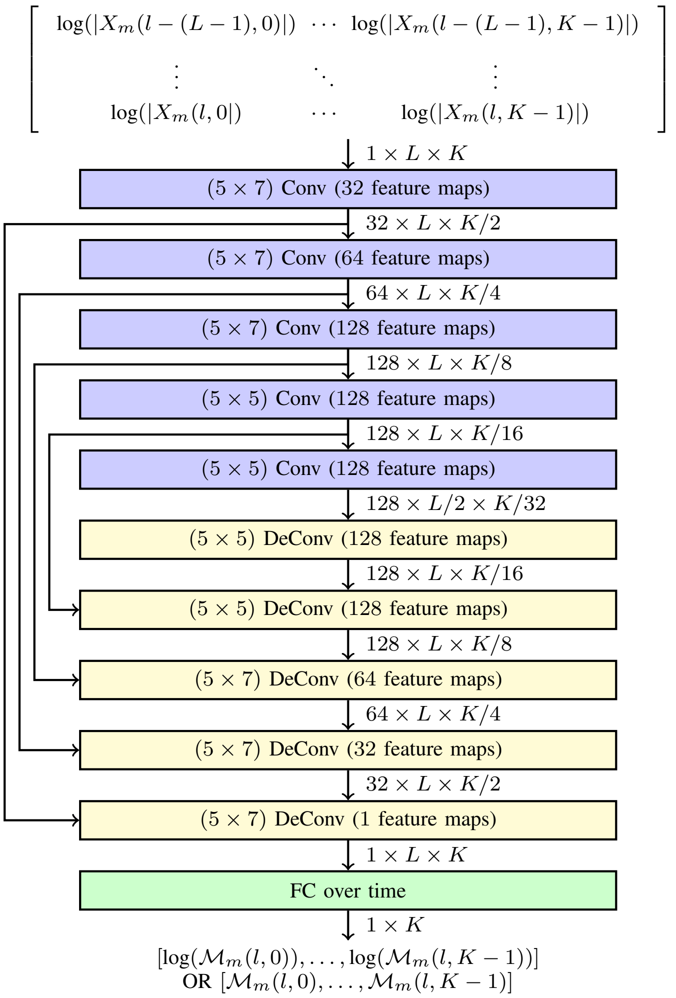

log(|X_m(l − (L − 1), 0|) · · · log(|X_m(l − (L − 1), K − 1)|)
log(|X_m(l, 0|) · · · log(|X_m(l, K − 1)|)
1 × L × K
(5 × 7) Conv (32 feature maps)
32 × L × K/2
(5 × 7) Conv (64 feature maps)
64 × L × K/4
(5 × 7) Conv (128 feature maps)
128 × L × K/8
(5 × 5) Conv (128 feature maps)
128 × L × K/16
(5 × 5) Conv (128 feature maps)
128 × L/2 × K/32
(5 × 5) DeConv (128 feature maps)
128 × L × K/16
(5 × 5) DeConv (128 feature maps)
128 × L × K/8
(5 × 7) DeConv (64 feature maps)
64 × L × K/4
(5 × 7) DeConv (32 feature maps)
32 × L × K/2
(5 × 7) DeConv (1 feature maps)
1 × L × K
FC over time
1 × K
[log(M_m(l, 0)), . . . , log(M_m(l, K − 1))]
OR [M_m(l, 0), . . . , M_m(l, K − 1)]

Fig. 1: UNet structure: convolution layers reduce the frequency dimension, while increasing the feature dimension. Deconvolution layers do the reverse. Skip connections are concatenated along the feature dimension before deconvolution. The features are then combined over time via a fully connected (FC) layer.

When SNR is low, IRMs of most time-frequency bins have a low value. Training in such conditions, the network has difficulties in learning the target because the dynamic range of the error is very small. One solution to this numerical issue is to apply a log compression to the training target as in [17], allowing the network to better generalise to low-valued IRMs.

Thus, two networks are trained, one having the standard IRM as training target and the other: the log-compressed IRM. To obtain the log-IRM training target, the standard IRM is transformed into dB scale and clipped between $-40\,\text{dB}$ and $0\,\text{dB}$ to avoid potential numercial issues from extreme values. The final output layers of the two networks needs to be modified accordingly as well. For standard IRM target (termed as linear IRM in the following), the FC layer output is rectified by the sigmoid function. For log IRM target, the output of the FC layer is only clipped between -40 and $3\,\text{dB}$. Other than that, the two networks share the same architecture. To ensure the convergence of the network training, batch-normalization is applied to each layer. Since the network is supposed to predict the IRM of the last frame in the input sequence, the zero-padding on the encoder side is one-sided (along the first frames). The transpose deconvolution is chosen for the decoder. For each frame, we utilise 15 previous frames to provide the contextual information to the networks, yielding $L = 16$ consecutive frames as input. All the parameters of the autoencoders are listed in Table. I.

TABLE I: Parameters of the autoencoder. Since encoder and decoder have symmetric structures, only the encoder structure is enumerated.

Encoder parameters	
Channels	32, 64, 128, 128, 128
Kernel size	(5,7), (5,7), (5,7), (5,5), (5,5)
Stride	(1,2), (1,2), (1,2), (1,2), (2,2)
Default activation functions	Leaky ReLU, slope= 0.03
Loss function	Mean square error (MSE)
Number of epochs	5

C. Adaptive beamforming

The application of the beamformer, with adaptive weights $\mathbf{W}(l,k)$, can be written as:

$$Y(l,k) = \mathbf{W}^H(l,k)\mathbf{X}(l,k)\,, \tag{7}$$

where $Y(l,k)$ represents the beamformer output.

We consider two beamformers: the adaptive minimum variance distortionless response (MVDR) beamformer and the adaptive multi-channel Wiener filter (MWF). Both approaches make use of spatial covariance matrices (SCM) to compute the weights. We denote the speech SCM as $\mathbf{\Phi}_{ss}$, the noise SCM as $\mathbf{\Phi}_{vv}$ and the input signal SCM as $\mathbf{\Phi}_{xx}$. The beamformer weights are then defined as [9]–[12]:

$$\mathbf{W}^{\text{MVDR}}(l,k) = \frac{\mathbf{\Phi}_{vv}^{-1}(l,k)\mathbf{\Phi}_{ss}(l,k)\,\mathbf{e}_{\boldsymbol{\ell}}}{\text{Tr}(\mathbf{\Phi}_{vv}^{-1}(l,k)\mathbf{\Phi}_{ss}(l,k))}\,, \tag{8}$$

$$\mathbf{W}^{\text{MWF}}(l,k) = \mathbf{\Phi}_{xx}^{-1}(l,k)\mathbf{\Phi}_{ss}(l,k)\,\mathbf{e}_{\boldsymbol{\ell}}\,, \tag{9}$$

where $\text{Tr}(.)$ is the trace operator and $\mathbf{e}_{\boldsymbol{\ell}}$ is a column vector where the ℓ-th element is one and the others are zeros. Setting ℓ to 1, the above expressions ideally try to reconstruct the speech component at the first microphone.

The true speech and noise spatial covariance matrices are not straight forward to obtain. Estimates are acquired with the help of masks generated by the AE, which act as *proxies* for the speech presence probability. Thus $\mathcal{M}(l,k)$ indicates the weights given to time-frequency bins when estimating $\mathbf{\Phi}_{ss}$, and the inverted mask $(\mathcal{M}_v(l,k) = 1 - \mathcal{M}(l,k))$ is used to estimate $\mathbf{\Phi}_{vv}$. This leads to the following weighted recursive averaging [14] on $\mathbf{X}(l,k)$ for estimating the SCMs:

$$\hat{\mathbf{\Phi}}_{ss}(l) = \alpha\hat{\mathbf{\Phi}}_{ss}(l-1) + (1-\alpha)\mathcal{M}(l)\mathbf{X}(l)\mathbf{X}^H(l) \tag{10}$$

$$\hat{\mathbf{\Phi}}_{vv}(l) = \alpha\hat{\mathbf{\Phi}}_{vv}(l-1) + (1-\alpha)\mathcal{M}_v(l)\mathbf{X}(l)\mathbf{X}^H(l) \tag{11}$$

$$\hat{\mathbf{\Phi}}_{xx}(l) = \alpha\hat{\mathbf{\Phi}}_{ss}(l-1) + (1-\alpha)\mathbf{X}(l)\mathbf{X}^H(l)\,, \tag{12}$$

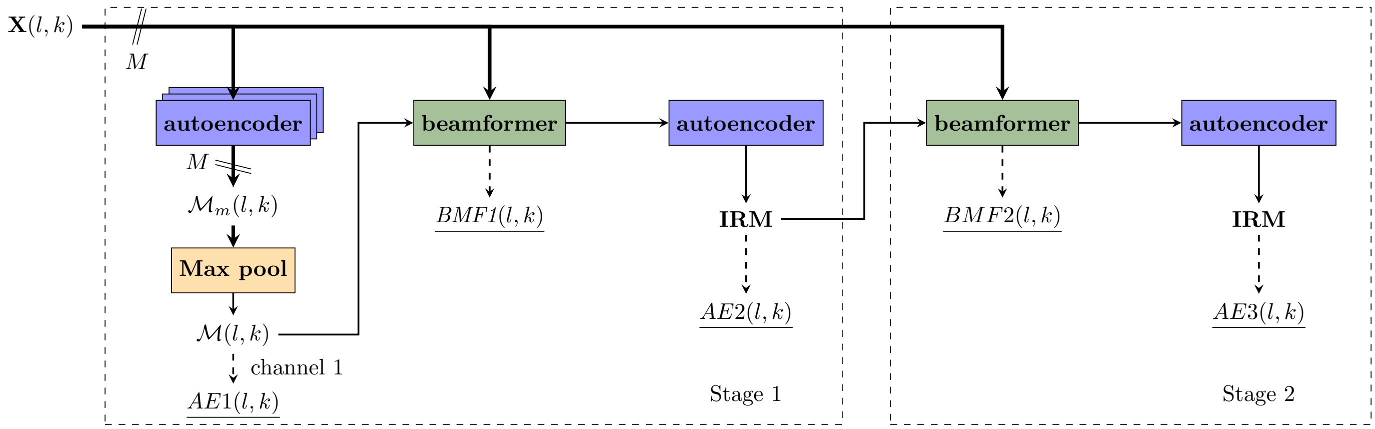


Fig. 2: Schematic of the proposed multistage beamformer. Thick lines indicate multi-microphone inputs and thin lines indicate the processing is single-channel. The same autoencoder and beamforming logic is used in all stages.

where α is the fixed averaging factor of the recursive method. Note: the averaging is performed independently at each frequency k, hence we omit frequency index for conciseness in the above. In practice, the beamformers are then implemented in a similar manner to eqs. (8) and (9), but with the estimated SCMs taking the place of the real SCMs.

D. *Multistage Beamformer*

A good estimate of the beamformer coefficients is depedent on obtaining correct statistics for the SCMs, which in turn requires a good mask estimate from the AE. The initial mask estimate, from the noisy microphone inputs, is somewhat poor due to the adverse input SNRs. In order to have a better mask estimation, the AE can be applied to the beamformed signal, which should have higher SNR than all the individual microphone signals. This improved mask can then be used to re-estimate beamformer coefficients, leading to an improved beamformer output, and this iterative process can be repeated multiple times. We term this *multistage beamforming*. This is an extra step compared to the current state of the art in drone noise suppression. We note that a similar approach has previously been reported to work well in speech recognition [18].

E. *Summary of method*

A schematic for the whole multistage beamformer process is shown in Fig. 2. The first stage consists of applying the AE to all M channels separately. The masks are then pooled in order to have a more reliable estimate. The pooled mask can either be applied directly to one of the channels of the microphone array, or be used in one of the beamformer algorithms to estimate the needed SCMs. The output of a direct application of the pooled mask to the reference microphone is denoted as AE1. The output from the first round of mask-steered beamforming will be denoted as BMF1.

To further suppress residual noise, a postfilter can be applied to the output of the MVDR beamformer. The postfilter is obtained by passing the beamformed output through the same AE. Applying this mask to the beamformed signal, the output is denoted AE2.

Since this mask should have a better estimate of where the speech and noise are situated, this can be used to re-estimate improved SCMs. Using these for the MVDR and MWF beamformers forms the second stage. The output is then denoted as BMF2. Similarly, a postfilter obtained by passing BMF2 through the AE can be applied to the signal, leading to the output signal AE3.

III. Evaluations

The proposed method is now evaluated. The baseline is the multi-channel Wiener beamformer (MWF), which includes both beamforming and post-processing. The intermediate results of the multistage beamformer are also investigated for the ablation analysis.

Three widely-used instrumental metrics are applied for the evaluation: the wide-band Perceptual Evaluation of Speech Quality - mean opinion score, listening quality objective (PESQ-mos-lqo, shorthand PESQ) [19], short-term objective intelligibility (STOI) [20] and the segmental signal-to-noise Ratio (segSNR). Since beamforming tries to reconstruct the speech component at the first microphone, the oracle speech component $s_1(n)$ of the first microphone is chosen as the reference signal for the intrusive metrics. Denoting the processed signal by $y(n)$, segSNR is defined as [2]:

$$\text{segSNR} = 10\log_{10}\left(\frac{\sum_l F(l)\,\frac{E_{s_1}(l)}{E_e(l)}}{\sum_l F(l)}\right),\ F(l) \in \{0, 1\} \tag{13}$$

where the error signal $e_k(n)$ is given by $e(n) = y(n) - s_1(n)$ and $E_{s_1}(l)$ and $E_e(l)$ represent the energies of the respective signals $s_1(n)$ and $e(n)$ in frame l, and $F(l)$ is a binary flag indicating speech presence/absence in the frame - which is obtained by an energy-based, oracle voice activity detector on the clean speech component.

A. *Database*

For training and validating our system we require multi-microphone databases of drone captures and there are a few available, e.g. the AIRA-UAS [21] dataset, the DREGON [22] dataset and the AVQ [4] dataset. However, AIRA-UAS and AVQ were recorded in outdoor situations. Thus, background noise is present in the clean speech references. Noise in the reference recordings would degrade the evaluation metrics and impede a proper analysis.

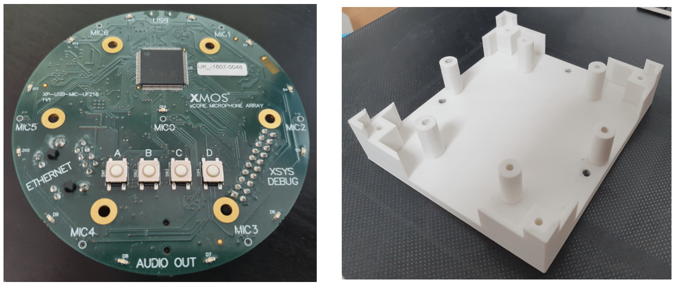
The DREGON dataset is recorded in two rooms with low reverberation times. However, here the speech is *narrowband*. Since we wanted to train our network on *wideband* speech samples, evaluation on DREGON would have a big mismatch with the training data. For these reasons we recorded our own multi-channel drone dataset, described in Section III-B.

For the training of single-channel AE, TIMIT [23] training set and part of our drone noise recordings are mixed at SNRs from $-20\,\mathrm{dB}$ to $0\,\mathrm{dB}$ with step $5\,\mathrm{dB}$. This yields in $18.9\,\mathrm{h}$ of training data in total. For the evaluation, drone recordings not seen during training are added to the speech recordings at SNRs in the same range.

B. Recording setup

The dataset of multichannel drone noise and clean speech signals was recorded in an semi-anechoic chamber. The semi-anechoic environment models typical drone situations well in the sense that during open-air use of drones, the only reflections are from the ground. In this setting, clean reference is easily available for the evaluation.

The microphone array used was a 7-channel xCore circular array (Fig. 3a), fixed by a 3D-printed fixture (Fig. 3b) directly to the bottom of the drone (3DR Solo quadcopter). Other datasets were recorded either with the arrays further below or above the drone with additional mechanical structures. In contrast, our setup makes it possible for a more compact drone.



(a) xCore microphone array

(b) Printed base

Fig. 3: The microphone array and its fixture

The recording setup is shown in Fig 4. The vertical distance h between the speaker and the array was fixed to $1.16\,\mathrm{m}$ for all the recordings. During the measurements, the drone was anchored in place as shown, allowing us to simulate various flying modes - which makes for a controllable scenario.

The details of the recording are shown in Table. II. Two scenarios were simulated: drone noise only (R), where the drone flew either in a stable mode or in a dynamic way, and speaker only (S). Two locations were considered during the measurement for the speaker, namely, directly under the drone ($d = 0\,\mathrm{m}$), or in front of the drone ($d = 2\,\mathrm{m}$). In each position, 100 clean utterances from TIMIT were played back by the loudspeaker and recorded by the array.

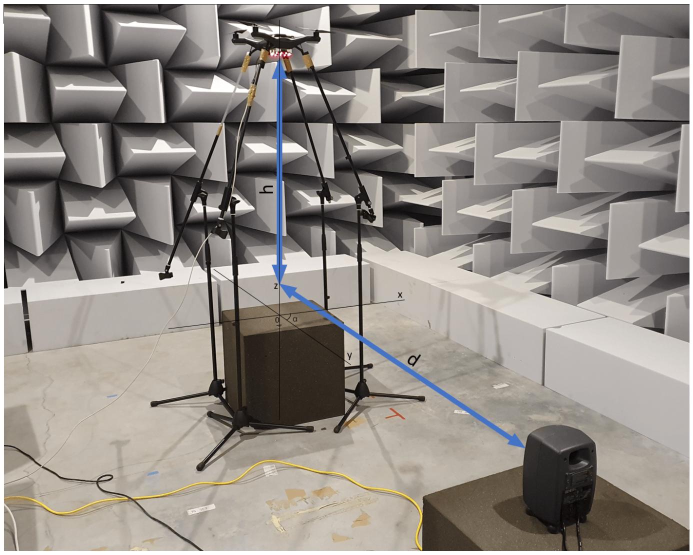


Fig. 4: Recording setup in the semi-anechoic chamber. The drone was fixed on the ground by four tripods, and clean utterances were played by the speaker. $h = 1.16\,\mathrm{m}$, $d = 2.0\,\mathrm{m}$

TABLE II: Drone noise dataset: specifications

	d (m)	Drone	Duration (sec)
R1	-	stable	648
R2	-	dynamic	294
S1	0	-	309
S2	2	-	300

C. Methods

All the signals are downsampled to $16\,\mathrm{kHz}$ before further processing. Signals are segmented into frames of $N (= 2K) = 512$ samples with 50% overlap, and windowed by a square-root von Hann window before transforming into the frequency domain. The (log-)amplitude spectrum of the positive frequencies is then input to the AE. To maintain the stride along the frequency, the highest frequency in the signal spectrum is trimmed as input. As a result, the IRM for the highest frequency bin is set to 0. This is not a problem since this bin corresponds to the Nyquist frequency.

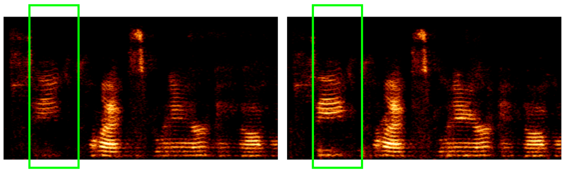
The multi-channel Wiener filter is taken as a baseline for the proposed system. Apart from evaluating the final output of the multistage approach, all the underlined intermediate results in Fig. 2 are also investigated for an ablation analysis.

D. Results and discussions

1) Comparison of pooling schemes: First we will discuss our findings on different mask pooling options. Generally, average or median pooling is taken as the default pooling layer. In [18], median pooling was reported to work best for multi-microphone speech enhancement in conjunction with automatic speech recognition (ASR). But the conditions are vastly different to the case of drone recordings - the SNR in [18] is much higher (on average) and the interference is more sparse (temporally and spectrally). In table III, the difference between median and max pooling after first stage beamforming (BMF1) and after postfiltering the first stage beamformed output (AE2) is shown. The metrics indicate the *improvement* relative to the noisy input, averaged over different SNRs. These output stages are chosen since they are central for multistage beamforming and analysing the performance differences here would help understand the effect for the succeeding stages.

TABLE III: Comparison of the improvement from the noisy input by max-pooling and median-pooling for the first stage (AE trained on linear IRM target).

method	max pooling		median pool	
	BMF1	AE2	BMF1	AE2
ΔPESQ	0.089	0.135	0.061	0.086
ΔSTOI	0.052	0.032	0.037	0.008
ΔsegSNR [dB]	8.03	8.70	8.16	8.70



(a) median pooling

(b) max pooling

Fig. 5: Comparison between median and max pooling after first stage MVDR beamforming and AE postfiltering (AE2)

Firstly, it is clear that max-pooling would, in comparison to other methods, mask fewer time-frequency points. Thus when applied directly, it would result in more speech, but also more noise in the output. These masks are, however, used to steer the adaptive beamformer. From table III we see that when using max-pooled masks for beamformer steering, the PESQ and STOI metrics are higher than when using median-pooled masks. This indicates that the resultant beamformer and AE postfilter preserve more speech. The trade-off is lower noise suppression (lower ΔsegSNR). However, when listening to audio examples, the speech is clearer with max pooling – thus supporting the perceptual and intelligibility metrics. This can also be appreciated in Fig. 5, where the better speech preservation is clearly visible (a section is highlighted for convenience). The trend is consistent on all test files.

Thus we conclude that a mask which lets through more speech at the cost of less noise suppression is more effective in steering the beamformer than a more aggressive mask.

2) Training target comparison: The effect of the training target on the autoencoder is shown in Table IV, where we break down the output of the first autoencoder, 'AE1', by input SNRs. From STOI, we see that the denoised signals have acceptable speech intelligibility (STOI > 0.6) when the input SNR is ≥ -5dB for both AEs, which is already quite ideal in the application of drone noise recording. ΔsegSNR indicates that log-compressed IRM provides a better noise suppression than standard IRM in all conditions, which also brings a benefit in speech quality measured by PESQ. The cost of this stronger suppression is slightly more distortion in speech, which is reflected by a lower STOI.

TABLE IV: Comparison of IRM and log-IRM as the training targets for the AE. Evaluation is performed on the direct output of AE. Training on log-IRM suppresses more noise, but introduces more speech distortion.

Input SNR	-20dB	-15dB	-10dB	-5dB	0dB
AE trained on linear IRM target					
PESQ	1.080	1.047	1.062	1.122	1.253
STOI	0.355	0.436	0.535	0.628	0.721
ΔsegSNR [dB]	8.56	7.45	6.55	5.41	4.45
AE trained on log-IRM target					
PESQ	1.081	1.054	1.072	1.135	1.253
STOI	0.341	0.420	0.522	0.620	0.713
ΔsegSNR [dB]	11.14	9.80	7.85	5.87	3.55

To select the best network to predict the mask for the beamformer, we compared the performance of the second and the third autoencoder output ('AE2' and 'AE3') in table V. Interestingly, the AE trained on the linear IRM improves the cascaded system performance as expected but the AE trained on a log-IRM target *degrades* the output speech when it is applied in cascade.

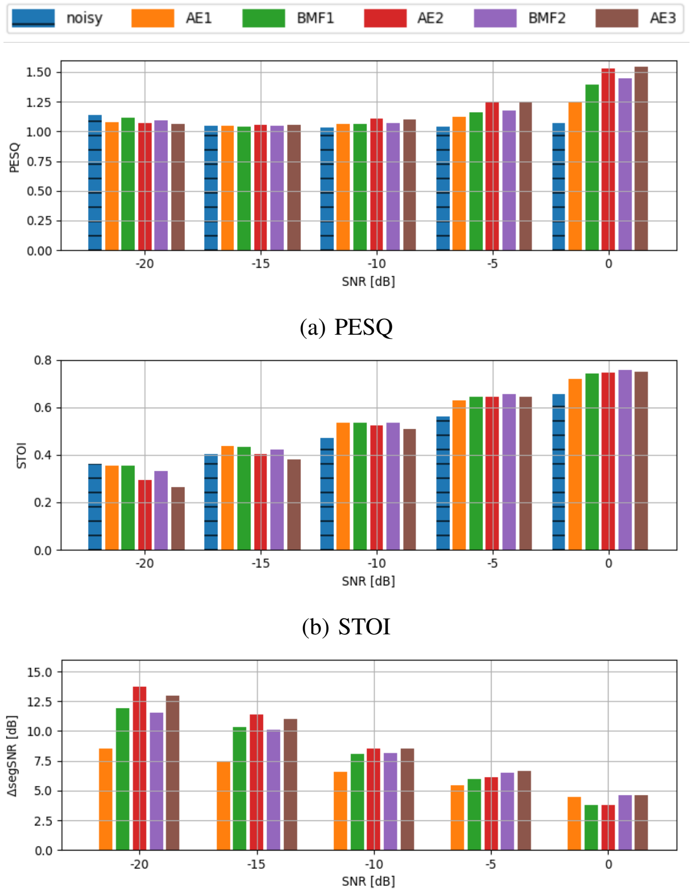
TABLE V: Comparison of training targets when used in combination with multistage beamforming.

method	linear IRM		logarithmic IRM	
	AE2	AE3	AE2	AE3
ΔPESQ	0.14	0.13	0.09	0.07
ΔSTOI	0.03	0.02	-0.01	-0.03
ΔsegSNR [dB]	8.67	8.75	8.20	7.87

3) Benefit of multistage processing: The results of the previous sections indicate that max-pooling of the masks and using an AE trained on the linear IRM target are best at predicting the masks for the subsequent beamformer. In Fig. 6, we evaluate the benefit of the multistage combination. Since this constitutes the final system, we present the results more elaborately by grouping along the input SNRs. As may be seen, for extremely adverse conditions, although segSNR improves a lot, there is no gain in terms of speech quality or speech intelligibility. The proposed system starts yielding a consistent and reasonable improvement in terms of intelligibility and quality from -10 dB onwards. However, the multistage combination only starts to become beneficial when the input SNR is high enough (SNR≥ -5 dB).

In contrast, the multistage beamformer using an AE trained on the log-IRM target demonstrates a different tendency along the processing chain (Fig. 7). Here, successive stages actually *degrade* the output signal in all three metrics. To get an intuition behind this behaviour, we plotted the evolution of the spectrogram of a -10 dB sample in Fig. 8. It can be observed from this set of samples that the weak speech harmonics gradually lose its shape along the processing while more noise is suppressed. This observation explains the degradation of the multistage beamformer when using log-IRM autoencoder.

4) AE as a postfilter: We now compare the benefit of using AE to predict the mask as a postfilter against the baseline multichannel Wiener filter approach (which implicitly combines



(a) PESQ

(b) STOI

(c) ΔsegSNR

Fig. 6: Evaluation of the system based on an AE trained on a linear-IRM target. Results are grouped by input SNR. The basic systems (BMF1, AE2) yield a reasonable output when SNR≥ -10dB. The multistage processing starts providing additional benefit when SNR≥ -5dB

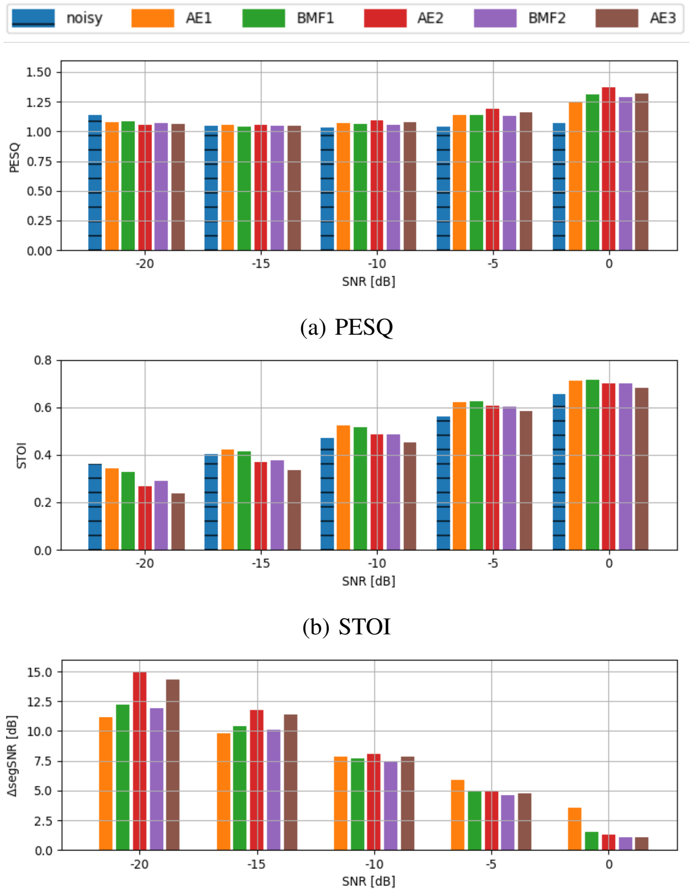
beamforming and postfiltering). The AE mask is computed on, and applied to the beamformer output BMF1 (yielding, thus, the output AE2 in our schematic). The linear-IRM-trained AE system was used. The comparison is presented in Tab. VI. These results show that using the AE to estimate a postfilter mask on the beamformed output is a viable technique and surpasses the classical Wiener filter as the post processing method.

TABLE VI: Benefit of an AE-estimated postfilter compared to the classical MWF. Metrics are averaged on the whole testset.

method	BMF1	AE2	MWF
ΔPESQ	0.09	0.14	0.08
ΔSTOI	0.05	0.03	0.00
ΔsegSNR [dB]	8.03	8.70	6.56

IV. Conclusions

We proposed a multistage beamformer scheme to enhance audio captured by a drone-mounted microphone array. A single-channel autoencoder (AE) plays a key role in the system. The AE predicts an ideal ratio mask (IRM) which is used to steer the beamformer. In successive stages, the beamformer output is fed into the same AE to obtain improved masks, which are then iteratively used to re-focus the beamformer. The results show that speech quality and intelligibility can be improved by this multistage system when SNR≥ -5 dB. As training target of the AE, we compare the IRM and log-compressed IRM. Although log-IRM demonstrates better performance in the first stage (AE-estimated mask applied to the microphone signal), it is too aggressive to be a preferable choice for the multistage system. The strong noise suppression diminishes the weak speech components as well, which cannot be recuperated in successive stages. We also verified that median pool, as the most common pooling method to generate a pooled mask from a multi-channel signal, is not the best choice for our system. Max-pooling, which preserves more speech components at the cost of decreased noise suppression, helps in a more effective steering of the beamformer. We note that while we addressed several interesting issues, there is still a mismatch between real recordings and the synthetic

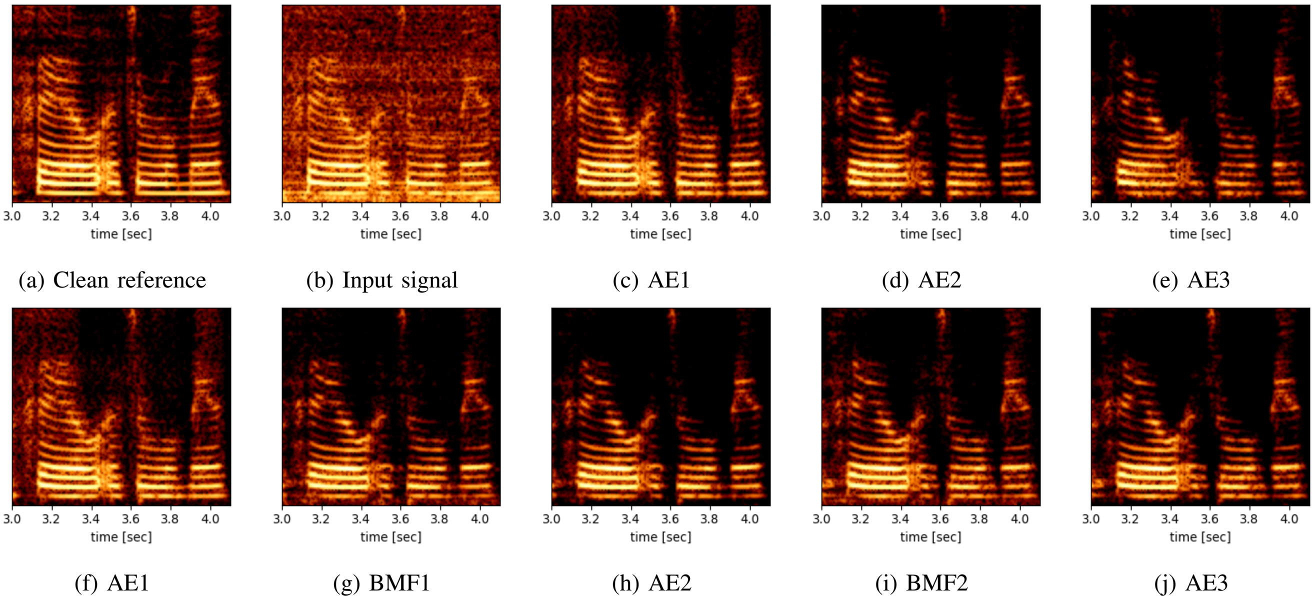


(a) PESQ

(b) STOI

(c) ΔsegSNR

Fig. 7: Evaluation of the system using AE trained on a log-IRM target. In contrast to Fig. 6, no clear improvement tendency can be observed. Additional stages successively *degrade* the output. The overall results, even for the earlier stages, are worse compared to Fig. 6.



(a) Clean reference (b) Input signal (c) AE1 (d) AE2 (e) AE3

(f) AE1 (g) BMF1 (h) AE2 (i) BMF2 (j) AE3

Fig. 8: Evolution of spectrogram in the multistage beamformer based on different AEs, using the same input at SNR $= -5\,\text{dB}$. Upper panel: references and stage outputs of log-IRM-trained AE; lower panel: stage outputs of linear-IRM-trained AE.

data which mix pure drone noise and clean speech. In real recordings the rotor-generated turbulence will interfere with the sound wave – which cannot be simulated from the current database. In order to train systems for such cases, we need a well-labeled database with such realistic recordings. In parallel, we shall explore if we can mitigate the influence of this turbulence by e.g., a mechanical shield around the array.

References

[1] Y. Hioka, M. Kingan, G. Schmid, and K. A. Stol, "Speech enhancement using a microphone array mounted on an unmanned aerial vehicle," in *Intl. Workshop on Acoustic Signal Enhancement (IWAENC)*, 2016.

[2] L. Wang and A. Cavallaro, "Acoustic sensing from a multi-rotor drone," *IEEE Sensors Journal*, vol. 18, no. 11, pp. 4570–4582, 2018.

[3] D. Salvati, C. Drioli, G. Ferrin, and G. L. Foresti, "Beamforming-based acoustic source localization and enhancement for multirotor uavs," in *European Signal Processing Conf. (EUSIPCO)*, 2018, pp. 987–991.

[4] L. Wang, R. Sanchez-Matilla, and A. Cavallaro, "Audio-visual sensing from a quadcopter: dataset and baselines for source localization and sound enhancement," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, 2019, pp. 5320–5325.

[5] L. Wang and A. Cavallaro, "A blind source separation framework for ego-noise reduction on multi-rotor drones," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 28, pp. 2523–2537, 2020.

[6] ——, "Deep learning assisted time-frequency processing for speech enhancement on drones," *IEEE Trans. on Emerging Topics in Computational Intelligence*, vol. 5, no. 6, pp. 871–881, 2020.

[7] S. Yoon, S. Park, Y. Eom, and S. Yoo, "Advanced sound capturing method with adaptive noise reduction system for broadcasting multicopters," in *IEEE Intl. Conf. on Consumer Electronics*, 2015, pp. 26–29.

[8] S. Yoon, S. Park, and S. Yoo, "Two-stage adaptive noise reduction system for broadcasting multicopters," in *IEEE Intl. Conf. on Consumer Electronics (ICCE)*, 2016, pp. 219–222.

[9] E. A. Habets, J. Benesty, S. Gannot, and I. Cohen, "The MVDR beamformer for speech enhancement," in *Speech processing in modern communication*. Springer, 2010, pp. 225–254.

[10] A. Spriet, M. Moonen, and J. Wouters, "Spatially pre-processed speech distortion weighted multi-channel wiener filtering for noise reduction," *Signal Processing*, vol. 84, no. 12, pp. 2367–2387, 2004.

[11] N. Madhu, *Acoustic source localization: Algorithms, applications and extensions to source separation*. Der Andere Verlag, 2010.

[12] M. Souden, J. Benesty, and S. Affes, "On optimal frequency-domain multichannel linear filtering for noise reduction," *IEEE Trans. on audio, speech, and language processing*, vol. 18, no. 2, pp. 260–276, 2010.

[13] H. Erdogan, J. R. Hershey, S. Watanabe, M. I. Mandel, and J. Le Roux, "Improved MVDR beamforming using single-channel mask prediction networks." in *Proc. INTERSPEECH*, 2016, pp. 1981–1985.

[14] S. Chakrabarty and E. A. Habets, "Time–frequency masking based online multi-channel speech enhancement with convolutional recurrent neural networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 4, pp. 787–799, 2019.

[15] A. E. Bulut and K. Koishida, "Low-latency single channel speech enhancement using UNet convolutional neural networks," in *IEEE Intl. Conf. on Acoustics, Speech, Signal Processing*, 2020, pp. 6214–6218.

[16] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 26, no. 10, pp. 1702–1726, 2018.

[17] A. Bohlender, A. Spriet, W. Tirry, and N. Madhu, "Neural networks using full-band and subband spatial features for mask based source separation," in *European Signal Processing Conf. (EUSIPCO)*, 2021, pp. 346–350.

[18] X. Xiao, S. Zhao, D. L. Jones, E. S. Chng, and H. Li, "On time-frequency mask estimation for MVDR beamforming with application in robust speech recognition," in *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 3246–3250.

[19] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs," in *IEEE Intl. Conf. on acoustics, speech, and signal processing.*, vol. 2, 2001, pp. 749–752.

[20] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech," in *IEEE Intl. Conf. on acoustics, speech and signal processing*, 2010, pp. 4214–4217.

[21] O. Ruiz-Espitia, J. Martinez-Carranza, and C. Rascon, "AIRA-UAS: an evaluation corpus for audio processing in unmanned aerial system," in *Intl. Conf. on Unmanned Aircraft Systems*, 2018, pp. 836–845.

[22] M. Strauss, P. Mordel, V. Miguet, and A. Deleforge, "DREGON: Dataset and methods for UAV-embedded sound source localization," in *IEEE/RSJ Intl. Conf. Intelligent Robots and Systems*, 2018, pp. 1–8.

[23] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1," *NASA STI/Recon technical report n*, vol. 93, p. 27403, 1993.