FACULTY OF ENGINEERING
AND ARCHITECTURE

Towards Deep Localisation and Separation with Ad Hoc Distributed Microphone Arrays

Stijn Kindt

Doctoral dissertation submitted to obtain the academic degree of
Doctor of Electrical Engineering

Supervisors
Prof. Nilesh Madhu, PhD - Prof. Tijl De Bie, PhD
Department of Electronics and Information Systems
Faculty of Engineering and Architecture, Ghent University

March 2025

GHENT
UNIVERSITY

ISBN 978-94-6355-968-3
NUR 962
Wettelijk depot: D/2025/10.500/28

Members of the Examination Board

Chair

Prof. Sabine Wittevrongel, PhD, Ghent University

Other members entitled to vote

Prof. Jeroen Hoebeke, PhD, Ghent University
Prof. Jesper Rindom Jensen, PhD, Aalborg University, Denmark
Prof. Hong-Goo Kang, PhD, Yonsei University, South Korea
Pieter Thomas, PhD, Ghent University

Supervisors

Prof. Nilesh Madhu, PhD, Ghent University
Prof. Tijl De Bie, PhD, Ghent University

Acknowledgments

First and foremost, I want to thank my promoter, Prof. Nilesh Madhu. Your guidance has had the most direct impact on my research career, transforming me from a novice to someone capable of writing this PhD thesis. You taught me both hard and soft skills and lent me your many connections. Most of all, I appreciate you giving me the opportunity to work on research topics that genuinely excite me.

I also want to thank Prof. Tijl De Bie for acting as my co-supervisor. Although our interactions were not numerous, your perspective was always valuable.

To the examination board: your feedback on this thesis has been invaluable, and I hope you agree that it has improved because of it. Your time and energy in reviewing this thesis and participating in the defences are truly appreciated.

Daily PhD life would not be as enjoyable without the people around the office. It is great having all of you around. Besides the laughs, evening events, and conferences we attended together, there is also a collaborative atmosphere that I cherish. Specifically, I want to thank Alexander; collaborating with you during the first period of my PhD helped me tremendously.

During my PhD, I not only collaborated with my team members but also had many opportunities to work with members from other universities. I want to thank Luca Becker and Prof. Reiner Martin from Ruhr University Bochum, Jihyun Kim and Prof. Hong-Goo Kang from Yonsei University, and Shuai Tao and Prof. Jesper Rindom Jensen from Aalborg University for past, current, and hopefully future collaborations.

I also enjoyed the collaborations with companies, which made the research more grounded and steered the research questions towards designing systems that could eventually be deployed. Here, all the members of the imec.Icon BLE2AV project have greatly contributed.

Similarly, my six-month internship at Elevear is also very much appreciated. I truly experienced what it is like to be at the intersection of research and valorization and to make real-world applications. I want to thank Johannes Fabry and Stefan Liebich for giving me this opportunity, and thank every single member of the team for making it so enjoyable.

I also want to thank the many master's thesis students I had the honour to guide. Your work inspired many new ideas, some of which have even ended up in a paper.

Thanks are not only due to my work-related friends but also to the many people that make life outside of work a pleasure. There is the "peter en meter" group, which started in my first week at Ghent University and was meant to guide us through the first year of college. However, the group became so much more, and I

am happy to still call all of you my friends. I still cherish our first ”kot” gatherings after the ”testjes”, where stacks of pizza boxes were collected.

Later, by the innocent choice of selecting electrical engineering as a major, an extra group of friends joined. ”De elektro’s” became closer each year we joined in the same classes and endured the same projects together. But it did not stop there; we embarked on many great after school adventures.

I also want to thank all my other friends, with whom I go sporting, play board games and DnD, co-house, or generally meet for food and/or drinks.

My family also deserves a thank you. You encouraged me to always do my absolute best and made me feel safe and supported in every choice I made throughout my life.

Last but certainly not least, I want to thank Margje. Your support is truly remarkable. You have endured many of my late-night and weekend deadline crunches. In fact, you are enduring one right now as I am writing this text. Thank you for making this journey as enjoyable as it has been, through celebrating my successes, making absolutely great food, encouraging me to be better, and your general love. To many more ”vijfjes”!

Gent, March 2025
Stijn Kindt

Table of Contents

Summary

Speech is a crucial part of our daily lives. Effective communication is crucial for collaboration and socialisation. Therefore, it is not surprising that technologies have been invented to improve our daily interactions. Phones enable communication over long distances, virtual meeting platforms facilitate teamwork and hearing aids help people with hearing loss better understand their surroundings and conversational partners.

The ever-growing list of speech communication technologies have one thing in common: they all contain some digital signal processing (DSP) algorithms to extract and enhance the speech captured by one or multiple microphones on the device. In recent years, these DSP technologies have increasingly trended towards deep learning paradigms, enabled by advancements in hardware capabilities. Learning from relevant and diverse sets of training data makes the deep neural network models generalise better than their classical counterparts, which are based on signal models with simplified assumptions.

Nevertheless, the signal models have proven useful for many decades of speech enhancement, providing valuable insights into signal properties. Infusing the deep neural network (DNN) designs with this domain-specific information should therefore be beneficial. This infusion is one of the themes that run throughout the chapters. Another theme is that of different microphone setups, gradually moving towards more distributed setups.

A device can have one or multiple microphones; the latter configuration is known as a microphone array. Also, multiple devices could be carefully placed in a room and exchange information, forming distributed microphone array setups. Alternatively, all microphones present in everyday situations (*e.g.*, phones, laptops, earbuds, *etc.*) can be combined to form *ad-hoc* distributed microphone setups, or *ad-hoc* distributed microphone arrays if each device contains multiple microphones. The setups where multiple devices are connected are also called wireless acoustic sensor networks (WASNs).

Having access to multiple devices in a room can be highly beneficial, as the sound field is sampled at drastically different places, potentially being dominated by the different sources of interest. This spatial diversity can be exploited for improved speech enhancement and speaker separation. However, distributed microphone processing comes with its own set of challenges. The devices all run on their own clocks, which have a slight sample rate offset and are independent of each other. Also, the devices are connected only via a wireless link, typically limiting the bandwidth capacity and thus the information exchange. In case these

nodes are combined in an ad-hoc fashion instead of deliberately placed, neither the place nor number of nodes is known a priori. This work features separation or localisation with one single compact microphone array (Chapter 3), two distributed microphone arrays (Chapter 4) and ad-hoc distributed, individual microphones (Chapters 5 to 8). All these individual works, with the right adaptation, could in the future be combined for ad-hoc distributed microphone array processing.

Specifically, **Chapter 3** deals with the DNN based separation of multiple sources with one compact microphone array. The architecture builds on previous work that employs location aware outputs, with each output corresponding to a specific look direction. In this chapter, location based information of the active sources is provided as additional input to the network. This location information is essential in many classical source separation methods to steer spatial filters.

Two location features are tested. The hand-crafted "expected phase differences" for sources coming from a particular direction, and a multi hot encoding of the speaker locations, where the DNN learns its own representation. Both features improve the separation when the sources are closely spaced. However, the location aware output representation was already sufficiently spatially distinctive for widely spaced sources. Among the two feature representations, the multi hot vector proved superior, indicating that the DNN was able to come up with a better representation. Also, the expected phase differences are based on the far field assumptions, which might not always hold and degrade the performance.

In **Chapter 4**, the goal is to localise the sources in 2D coordinates using two distributed microphone arrays and DNNs. Co-operative approaches were proposed, where parts of the neural networks are shared. This significantly improved localisation accuracy compared to only sharing the direction of arrivals estimates, computed at each array independently. Two versions were tested: one that shared broadband information and the other shared narrowband information. From a bandwidth perspective, the broadband features are preferable, but they share less information between the arrays. Surprisingly, the broadband version also performs better, likely due to spatial aliasing effects at higher frequencies in the narrowband approach. Another notable result of this work is that the system is inherently robust against clock offsets since only features are shared between the arrays.

From Chapter 5 onwards, all chapters focus on ad-hoc distributed microphones. This field of research is relatively new, with limited prior research available. Exploiting their spatial diversity should be beneficial for speech separation and enhancement. Yet, dealing with unknown microphone numbers and positions renders conventional methods impractical. Especially DNN based methods, which are typically trained for a specific array configuration. To address this the microphones are clustered with respect to their dominant sound sources.

In **Chapter 5**, a new clustering feature is proposed: speaker embeddings, which are compared to another speaker-specific feature: modulated Mel frequency cepstral coefficients (Mod-MFCC). The evaluation was performed on a realistically simulated dataset, under varying difficulties: the sources were put close to each other, making it harder to identify which microphones belong to which cluster. Also, the time on which the features are computed is shortened, which is important

for dynamic situations. For all scenarios, the speaker embeddings worked well and outperformed Mod-MFCC features.

Chapter 6 then compares the speaker embedding features with coherence based clustering. The evaluations show that the coherence based clustering performs slightly better than the speaker embeddings. However, other trade-offs were identified. Firstly, the coherence based clustering needs all microphone signals to be present at a central node, which requires a much larger bandwidth than sending speaker embeddings. Additionally, speaker embeddings could be used to identify target clusters, based on a target embedding. On the other hand, the full signals are often needed for subsequent tasks like speaker separation, effectively sharing the bandwidth for clustering and separation, reducing the overall bandwidth.

Chapter 7 showcases that clustering is essential for DNN based separation with WASNs. If all microphones are inputted to an array agnostic separation DNN (a model that does not require any predefined array geometry), the model fails to learn the inter-microphone relations and the separation fails. However, the array agnostic models work wonderfully if only the microphones corresponding to a single cluster are provided, stressing the necessity for *informed* DNNs.

Additionally, **Chapter 8** further improves upon Chapter 7 with cross-cluster information exchange, something that is essential for the previous classical separation methods. This further improves the quality of the separated speech signals.

In the **future**, these systems could be brought together, with adequate adaptation, to perform deep localisation and separation with ad-hoc distributed microphone *arrays*. Each array can perform spatially selective extraction, in predefined look areas or beams. This will create virtual microphone signals, that contain the sources within these predefined beams. Then, the beams of all microphone arrays can be combined for co-operative speech separation and enhancement, improved by the added spatial diversity. Each virtual microphone signal can be used in the array agnostic enhancement frameworks of Chapters 7 and 8 as input. However, only signals which are dominated by the same source should be used together. Therefore, the virtual microphone signals can be clustered in the same fashion as the microphone signals in Chapters 4 and 5. Alternatively, co-operative localisation methods (Chapter 4) could be used to localise the sources and to inform which virtual microphone signals should be jointly processed. Once the exact source locations are known, an improved location informed separation (Chapter 3) can take place on each microphone array, improving the inputs of the array agnostic DNN.

Samenvatting
– Dutch Summary –

Spraak is een cruciaal onderdeel van ons dagelijks leven. Communicatie is essentieel voor goede samenwerkingen sociaal contact. Het is daarom niet verrassend dat er technologieën ontwikkeld worden om onze dagelijkse communicatie te verbeteren. Denk maar aan smartphones, online team meetings en hooraparaten.

De groeiende lijst aan communicatietechnologieën heeft eenzelfde achterliggende methode: door middel van digitale signaalverwerking die door één of meerdere microfoons op het apparaat wordt opgevangen, de spraak te isoleren en te filteren. In de afgelopen jaren zijn deze signaalverwerking technologieën steeds meer gebaseerd op deep learning paradigma's, mogelijk gemaakt door de steeds krachtiger wordende hardware. De modellen leren uit diverse trainingsdata, wat er voor zorgt dat de diepe neurale netwerkmodellen beter generaliseren dan hun klassieke tegenhangers. Die zijn namelijk veeleer gebaseerd op signaalmodellen met vereenvoudigde aannames.

Desalniettemin hebben die klassieke modellen hun nut afgelopen decennia en waardevolle inzichten en signaaleigenschappen opgeleverd. Deze domeinspecifieke informatie in het ontwerp van de diepe neurale netwerken (DNN) verwerken is daarom voordelig. Deze infusie is een van de thema's die doorheen de verschillende hoofdstukken loopt. Een tweede thema die doorheen deze thesis loopt is dat er verschillende microfoonopstellingen gebruikt worden.

Een apparaat kan één of meerdere microfoons hebben. Wanneer die er meerdere heeft, wordt dit ook wel een microfoonrooster genoemt. Ook kunnen meerdere apparaten in een kamer worden geplaatst en informatie uitwisselen, waardoor gedistribueerde microfoonrooster opstellingen ontstaan. Alternatief kunnen microfoons die in alledaagse situaties aanwezig zijn (bijv. GSMs, laptops, oortjes, ...) worden gecombineerd om ad-hoc gedistribueerde microfoonopstellingen te vormen, of ad-hoc gedistribueerde microfoonroosters als elk apparaat meerdere microfoons bevat. Deze laatste opstellingen worden ook wel draadloze akoestische sensornetwerken (DASNs) genoemd.

Het combineren van meerdere apparaten in één kamer kan heel voordelig zijn, aangezien het geluidsveld op heel verschillende plaatsen wordt bemonsterd, en mogelijk gedomineerd zijn door andere geluidsbronnen. Deze ruimtelijke diversiteit kan worden benut om de spraakverwerking te verbeteren. Echter, aangezien de microphoons zich niet langer op hetzelfde apparaat bevinden, brengt dat unieke uitdagingen met zich mee. De apparaten hebben allemaal hun eigen interne klok-

ken, die elk een licht andere bemonsteringsfrequentie hebben. Bovendien zijn de apparaten alleen draadloos verbonden, wat doorgaans een beperkte bandbreedte heeft en dus ook de informatieuitwisseling beperkt. In het geval dat deze apparaten ad-hoc worden verbonden met elkaar, zijn noch de plaats noch het aantal apparaten vooraf bekend. Tijdens dit werk zullen DNNs ontworpen worden die veschillende sprekers scheid en lokaliseert met één enkele compacte microfoonrooster (hoofdstuk 3), twee gedistribueerde microfoonroosters (hoofdstuk 4) of ad-hoc gedistribueerde, individuele microfoons (hoofdstukken 5 tot 8). Al deze afzonderlijke werken zouden in de toekomst, met de juiste aanpassing, gecombineerd kunnen worden voor ad-hoc gedistribueerde microfoonroosterverwerking.

Specifiek behandelt **hoofdstuk 3** de DNN-gebaseerde scheiding van meerdere bronnen met behulp van één compacte microfoonrooster. De architectuur bouwt voort op eerder werk dat locatie afhankelijke outputs gebruikt, waarbij elke output overeenkomt met een specifieke focusrichting. In dit hoofdstuk wordt de locatie informatie van de actieve bronnen als extra kennis aan het netwerk gegeven.

Twee versies van locatie-informatie worden getest. De handgemaakte 'verwachte faseverschillen' tussen microfoons voor bronnen die uit een bepaalde richting komen en een multi-hot vector van de sprekerlocaties, waarbij de DNN zijn eigen representatie leert. Beide methoden verbeteren het onderscheidingsalgoritme voor bronnen die zich dicht bij elkaar bevinden. Echter, de locatie afhankelijke outputrepresentatie was al krachtig genoeg voor ver uit elkaar liggende bronnen. Voor de bronnen die dicht bij elkaar liggen bleek de multi-hot vectorde superieure methode te zijn. Dit geeft aan dat de DNN in staat is om een betere representatie te bedenken dan de verwachte fase verschillen. Bovendien zijn deze gebaseerd op de aanname dat de bronnen zich ver genoeg van de microfoons bevinden, zodat de geluidsgolven als vlakke golven worden gezien.

Het doel in **hoofdstuk 4** is om geluidsbronnen te lokaliseren in 2D-coördinaten met behulp van twee gedistribueerde microfoonroosters en DNNs. Coöperatieve methoden werden voorgesteld, waarbij een deel van de neurale verwerking van de roosterinformatie gemeenschappelijk gebeurt. Dit verbeterde de lokalisatienauwkeurigheid aanzienlijk in vergelijking met het trianguleren van de onafhankelijk geschatte aankomsthoeken door elk rooster. Twee coöperatieve methoden werden getest: één die breedbandinformatie deelde en de andere die smalbandinformatie deelde. De methode die breedbandinformatie deelt, verstuurt minder informatie naar een centraal verwerkings punt, wat vanuit een bandbreedteperspectief zeker voordelig is. Verrassend genoeg presteert de breedbandversie beter, waarschijnlijk vanwege ruimtelijke aliasing bij hogere frequentiebanden van de smalbandinformatie. Een ander opmerkelijk resultaat van dit werk is dat het systeem inherent robuust is tegen klokafwijkingen. Dit komt doordat alleen latente variabelen uitgewisseld worden tussen de roosters, die niet meer afhankelijk zijn van het exacte bemonster tijdstip.

Vanaf hoofdstuk 5 richten alle hoofdstukken zich op ad-hoc gedistribueerde microfoons. Dit onderzoeksgebied is relatief nieuw waardoor er niet veel eerdere studies zijn gemaakt. Het benutten van ruimtelijke diversiteit zou voordelig moeten zijn voor spraakscheiding en -verbetering. Echter, de methoden moeten kunnen

omgaan met onbekende aantallen en posities van microfoons. Voor conventionele methoden is dit meestal niet het geval. Vooral DNN-gebaseerde methoden, die doorgaans zijn getraind voor een specifieke roosterconfiguratie, kunnen niet omgaan met deze ongekende microfoon posities. Om dit aan te pakken kunnen de microfoons eerst geclusterd worden naargalang welke geluidsbron dominant is bij welke microfoon.

In **hoofdstuk 5** wordt een nieuw kenmerk voorgesteld om te gebruiken in de clustering: sprekerembeddings. Dit kenmerk wordt vergeleken met een ander spreker-specifiek kenmerk: gemoduleerde Mel-frequentie cepstrale coëfficiënten (Mod-MFCC). De evaluatie werd uitgevoerd op een realistisch gesimuleerde dataset en met verschillende moeilijkheidsgraden: de bronnen werden dicht bij elkaar geplaatst, waardoor het moeilijker werd om te identificeren welke microfoons bij welke cluster horen. Ook werd de tijd waarop de kenmerken worden berekend, verkort, wat belangrijk is voor dynamische situaties. Voor alle scenario's werkten de sprekerembeddings goed en presteerden ze beter dan Mod-MFCC-kenmerken.

hoofdstuk 6 vergelijkt de sprekerembeddings met coherentie gebaseerde clustering. De evaluaties tonen aan dat de coherentie gebaseerde clustering iets beter presteert dan de sprekerembeddings. Echter zijn er ook nog andere afwegingen die in acht moeten genomen worden. Ten eerste heeft de coherentie gebaseerde clustering alle microfoonsignalen nodig bij een centraal verwerkings punt, wat een veel grotere bandbreedte vereist dan het verzenden van de sprekerembeddings. Bovendien kunnen sprekerembeddings worden gebruikt om te identificeren welke cluster de gewenste spreker bevat op basis van de embedding van de gewesnte spreker. Aan de andere kant zijn de volledige signalen vaak toch nodig, om taken zoals spraakscheiding te voltooien, waardoor de bandbreedte die voor clustering wordt gebruikt ook nuttig is voor deze andere taken.

hoofdstuk 7 toont aan dat clustering essentieel is voor het scheiden van spraaksignalen in DASNs met DNNs. Als alle microfoon signalen aan een rooster-agnostisch netwerk (een model dat geen vooraf gedefinieerde roostergeometrie vereist) worden gevoed, mislukt de scheiding falikant. Echter, de rooster-agnostische modellen werken uitstekend als alleen de microfoons die afkomstig zijn van dezelfde cluster worden gebruikt. Dit benadrukt de noodzaak voor *geïnformeerde* DNN's.

Daarnaast verbetert **hoofdstuk 8** het model van hoofdstuk 7 verder door informatie uit te wisselen tussen the verschillende clusters. Op die manier is er een referentie van hoe de interfererende bron eruit ziet. Dit is ook iets dat essentieel is voor de klassieke scheidingsmethoden en verbetert de kwaliteit van de gescheiden spraaksignalen.

In de **toekomst** zouden deze systemen, mits enkele aanpassing, samengebracht worden om diepe lokalisatie en scheiding uit te voeren met ad-hoc gedistribueerde microfoonroosters. Elk rooster kan ruimtelijk selectieve extractie uitvoeren van vooraf gedefinieerde focusgebieden door middel van bundelvorming. Dit creëert virtuele microfoonsignalen, die alle bronnen binnen deze vooraf gedefinieerde bundels bevatten. Vervolgens kunnen de bundels van alle microfoonroosters worden gecombineerd voor spraakscheiding en -verbetering met de toegevoegde ruimte-

lijke diversiteit gegeven door de verschillende roosters. Elk virtueel microfoonsignaal kan worden gebruikt in de roosteragnostische methoden van hoofdstukken 7 en 8. Echter, alleen signalen die worden gedomineerd door dezelfde bron mogen samen gebruikt worden voor optimale performantie. Daarom kunnen de virtuele microfoonsignalen eerst worden geclusterd op dezelfde manier als de microfoonsignalen in hoofdstukken 4 en 5. Alternatief kunnen coöperatieve lokalisatiemethoden (hoofdstuk 4) worden gebruikt om de bronnen te lokaliseren en te informeren welke virtuele microfoonsignalen gezamenlijk moeten worden verwerkt. Zodra de exacte bronlocaties bekend zijn, zou ook een verbeterde locatie-geïnformeerde scheiding (hoofdstuk 3) plaatsvinden op elke microfoonrooster, waardoor de invoer van de rooster-agnostische DNN wordt verbeterd.

1
Introduction

"One of our most important faculties is our ability to listen to, and follow, one speaker in the presence of others. This is such a common experience that we may take it for granted; we may call it "the cocktail party problem." No machine has been constructed to do just this, to filter out one conversation from a number jumbled together."

–Colin Cherry (1957)

Communication is one key ingredient that made humans the most dominant species on earth [1]. It enables us to share knowledge from generation to generation and work together more efficiently, facilitating constant improvements of our skills and tools, ultimately creating the civilisations and technologies we know today. Communication also enables social connections, which benefits our health and happiness [2].

Speech is one of the oldest forms of communication, far before writing. During that long time period, humans have become quite good at picking out the speech they are interested in. We are able to pick up spatial cues to know where the sound originates from, and then zone into that speech. Small time delays –interaural time difference (ITD) – and energy level differences – interaural level differences (ILD) – between our ears help guide that focus. The remarkable nature of this ability is showcased nicely at a cocktail party. Amidst various noise sources, music, and many surrounding groups engaged in conversations, you strive to concentrate on the person speaking within your circle. It is even more difficult if the party is set in a spacious room – perhaps a cathedral. The sound can bounce off the walls and

linger, creating an echo-like effect, called reverberation. These challenges have, appropriately, been termed the cocktail party problem [3].

For some people, hearing loss makes focusing on the speaker of interest even harder and more energy consuming [4]. In severe cases, individuals with hearing impairments may even avoid busy social activities altogether [5]. Luckily, our technological advancements can help here, with hearing aids able to boost the sounds and filter out the noise. A recent trend, waiting for its breakthrough, is the development of hearables designed to enhance the audio experience for people with normal hearing.

Besides aiding in face-to-face communication, technology has also enabled new forms of spoken communication, like phone calls and online meetings. Additionally, it is now useful to talk with your technological devices, like home or car assistance, where previously only rage prompted people to speak to their computers. For human-computer interaction, understanding the right command is highly dependent on how well the algorithm can isolate your voice. Asking "What does the weather look like", and receiving a detailed description of the leather's appearance would be far from ideal. Additionally, one might want live transcriptions to more easily follow an online video conference, requiring the computer to do the hard task of automatic speech recognition (ASR). Similarly, for human interactions, the voice should be well isolated. In a phone call, if your caller is in a noisy environment – perhaps a cocktail party – this could negatively impact your experience. This also holds for meetings, although loud ventilation and nearby footsteps would be a more common noise source.

One key denominator in any application where speech comprehension is necessary – whether by humans or computers – is ensuring that the target speech stands out above all other speakers and background noise sources. The field of audio signal processing helps to accomplish this. Signal processing paradigms utilise filters, often based on estimated statistics, to extract the target signal. Signal models are built and used to extract these statistics. However, these signal models are often based on assumptions and deploying these model based filters in a different environment might degrade the performance significantly. During the past years, a move towards deep neural networks (DNNs), big parametric models tuned by huge amounts of training examples, have proven to outperform classical statistic estimators. Instead of needing specific models for the estimators, the DNN learns how to extract the target speech directly from the training data. As long as the training data resembles the data in its targeted applications, the DNN can perform well and is said to generalise well to unseen conditions. This is a big advantage over classical models where the simplifying assumptions inevitably lead to a mismatch with the real data. Yet these classical signal processing models have brought some valuable insights into the properties of speech. Therefore, fusing the generalisable deep learning models with the knowledge the classical models provide, is one main

overarching focus of this dissertation.

The second theme of this dissertation is the variation in microphone setups across different devices. Different devices have a different number of microphones. For example, older or cheaper phones typically have a single microphone for recording and enhancing speech. In contrast, newer models often feature two or more microphones. One microphone is usually positioned closer to the mouth, while the other will pick up more of the background noise. This second microphone can therefore be used to better estimate the noise statistics while the first is used for the speech statistics. While having one microphone, both the noise and speech statistics should be estimated by the same microphone based on underlying models.

Multiple microphones are also present in home assistance devices. Such devices with an array of microphones are typically called microphone arrays. In contrast to the two microphones in the phone, all microphones pick up the target speech and noise at comparable energy levels. Nevertheless, spatial filters –termed beamformers – can be designed to extract sounds coming from specific directions and suppress sounds from other directions. These filters rely on the propagation delays the sound waves experience when moving through the air, inducing time delays on the microphone signals. Properly filtering all microphones individually and summing over them results in the desired beamformers.

Besides setups with one compact microphone array, one could also set up multiple microphones distributed in the room. If you have a big living area, you might want to set up multiple home assistance devices so at least one is close to you and can better understand your commands. The combined system could exploit both beamforming and the proximity of the devices towards different sound sources. These setups are termed distributed microphone arrays or acoustic sensor networks (ASNs). ASNs could also be utilised in hybrid meeting rooms, with for example one microphone array under the screen and another one in the middle of the table.

Unfortunately, the added spatial diversity does come at some cost. The devices are typically connected with a wireless link to a central processor, forming wireless acoustic sensor networks (WASNs). This requires additional bandwidth and power to transmit the audio signals. Moreover, the devices operate on slightly different clocks. Even minimal clock discrepancies will, over time, result in substantial differences. If the clock offsets are ignored, the filters will become misaligned, leading to unpredictable outputs. Consequently, clock realignment is a standard practice in WASNs.

Another common setup where a WASN can be recognised is in meetings. Many people bring their laptops and phones with built-in microphones to the meeting. Combining them would create an ad-hoc distributed microphone array. The ad-hoc nature indicates that the array configuration – the location and number of microphones – is unknown prior to deploying such setups. Settings with ad-hoc

distributed microphones are becoming more common as well due to the trend towards the Internet of Things (IoT), increasing the number of microphone-carrying devices (*e.g.*hearables, smartwatches and even smart fridges). For speech enhancement tasks, the distributed nature can again be useful to capture different sources and estimate their statistic. However, besides the clock differences, the unknown array configuration and the widely distributed nature make these settings considerably different to other setups.

The configurations mentioned above are just a few examples, and many other setups are possible. Regardless, different microphone setups have their own domain-specific signal models and considerations. This is how the two main themes of this dissertation – different microphone setups and the incorporation of domain-specific information into DNNs – come together.

The DNNs considered in this dissertation are speaker localisation and speaker separation. Both tasks are related in the sense that one could aid the other. If you know the location of a speaker, a beamformer can be steered towards them. Reversely, if the speech of one speaker is extracted, and little to no interference or background noise is present, the localisation accuracy will be higher. Nevertheless, first localising the speakers and then extracting them is the most common sequence.

1.1 Outline

Since this is a cumulative thesis, most chapters consist of one paper. Before each chapter, a short description frames the paper in relation to the common thread of the dissertation. Besides the paper chapters, the dissertation starts with this introductory chapter, where in Section 1.2 the contributions of each chapter and thus this dissertation will be laid out and in Section 1.3 a list of all publications I contributed to during my PhD is given, followed by some general background about the core concepts from this thesis in Chapter 2. Then, Chapters 3 to 8 contain the papers on which I was a main contributor (first or shared first author). Only one paper [6] is left out, to avoid unnecessary overlap with [7]. Lastly, Chapter 9 will conclude the dissertation by restating the main takeaways from each chapter. It will also discuss potential improvements left unexplored and showcase how all the works can be combined to achieve speech separation and localization with ad-hoc distributed microphone arrays.

1.2 Research contributions

The contributions of this dissertation can be divided in multiple ways, depending on the criteria chosen. The first split is based on the array configuration. This dissertation will cover the following three configurations: (i) one compact micro-

phone array (*comp.*), (ii) multiple distributed microphone arrays (*Distr.*) and (iii) ad-hoc distributed individual microphones (*Ad-hoc*). The second criterion is the DNN task. The two tasks are speaker localisation (*Loc.*) and speaker separation (*Sep.*). The last criterion for splitting the contributions is based on whether the chapter extracts (*Extr.*) domain-specific knowledge or utilises (*Util.*) the information in the DNN models. Table 1.1 gives this indication. This dissertation is ordered in increasing spatial diverse scenarios.

Table 1.1: An overview of the type of contributions associated with each chapter in this dissertation. A smaller check mark indicates that the task was also present in that chapter, but not its main focus.

	Array Type			**Task**		**Information**	
	Comp.	Distr.	Ad-hoc	Loc.	Sep.	Extr.	Util.
Chapter 3	✔				✔		✔
Chapter 4		✔		✔			✔
Chapter 5			✔	✔	✓	✔	
Chapter 6			✔	✔	✓	✔	
Chapter 7			✔		✔		✔
Chapter 8			✔		✔		✔

Chapter 3: Location informed deep speech separation.

In Chapter 3, one compact microphone array is utilised to separate two speakers with a DNN. There are separation DNNs that perform decently well where all microphone signals are used as input [8]. Spatial information is implicitly used by the neural network to distinguish between the different sources. In contrast, in classical beamforming methods, the locations of the speakers should be known (or estimated) in order to steer the beamformer towards this direction by compensating for the phase differences between the microphones that are expected for a sound source at that position.

The hypothesis therefore is that incorporating location information as input features would help the DNN to extract useful information. And if it is useful, which location based input feature is more optimal: is it better to include hand crafted phase difference features, which are used for beamformers, or let the neural network learn its own representation, based on provided location information?

Chapter 4: Source localisation with distributed microphone arrays.

This chapter is concerned with the localisation of sound sources with the help of multiple distributed microphone arrays. It extends a localisation DNN designed for one microphone array [9] to handle multiple arrays. One straightforward way would be to localise the sources at each array individually, and then combine the outputs. However, combining information from all arrays at an earlier stage, in-

stead of only their outputs, could improve the localisation accuracy. This hypothesis holds true in classical localisation methods [10] and is tested for DNNs in Chapter 4.

However, straightforwardly doing so by combining all microphone signals at the input would not account for the challenges associated with distributed microphone processing. The unpredictable and time varying (slight) sample rate offsets would make it hard for a DNN to learn the relation between the input features of the two arrays. Therefore, the proposed co-operative localisation algorithms process the microphone signals for each array independently at first, after which the features of the arrays are further processed together. A comparison of the localisation accuracies will be carried out with and without clock jitter to test its robustness towards this jitter. An additional reason for not processing the captured microphone signals from the two arrays simultaneously is spatial aliasing. When the wavelength of the sound waves is small compared to the distance between two microphones, the phase differences between the microphone signals no longer have a one-to-one relation with the direction from which the sound is coming. Another evaluation is consequently whether it is better to share narrowband information, which might contain more information but is prone to aliasing at higher frequencies, or broadband information.

Chapter 5 to Chapter 8: Cluster informed speaker separation with WASNs.

All these chapters are concerned with distributed microphones. The assumption is that each node consists of one microphone, which does not limit its usability if more are available (*e.g.*in modern phones). This field is a less explored area compared to single microphone and compact microphone array setups. The ad-hoc nature brings challenges of unknown microphone positions and unknown microphone numbers. To cope with all these unknowns, the microphones can be clustered around the sources of interest. This can be regarded as a type of localisation: even though the exact location of neither the source nor microphones is learned by clustering, good clustering provides relative localisation information that indicates which microphones are close to which sound source. This is useful for many subsequent tasks like source classification (*e.g.*[11, 12]) and separation (*e.g.*[13, 14]).

For the classical separation framework of [14, 15], the following information from the clustering is utilised: Microphones close to one source are dominated by that source, and vice versa. comparing information across *reference microphones* of the clusters enables the algorithm to identify the components belonging to each source and noise. This reference microphone can directly be given by soft clustering, where each microphone has a contribution to each cluster. The microphone with the highest contribution towards that cluster is then selected as the reference. In addition to using cross-cluster information, microphones within the same source

cluster can be combined for intra-cluster beamforming, with the necessary statistics derived from cross-cluster comparisons.

For all these algorithms, it is important that the clustering is reliable. Including microphones very far from the source of interest decreases the performance of the beamformer [13]. It becomes even more problematic when both clusters are dominated by the same source, as this means that comparing the clusters is essentially comparing the same underlying signal. This makes it very challenging to accurately estimate statistics for each source, which is necessary for creating effective spatial filters. Therefore, Chapters 5 and 6 will be concerned with the clustering itself. As part of these evaluations, a simple classical cluster based separation will be used to show the performance of different methods, therefore a small mark is given in Table 1.1.

These classical separation frameworks, although useful in indicating the cluster performance, leave room for improvement. In the last chapters, Chapters 7 and 8, the separation algorithms with DNNs will be proposed. Here the DNNs will be highly informed by the clustering algorithm, and a comparison with less informed architectures will be performed. The exact contributions of each particular chapter will be discussed below.

Chapter 5 deals with microphone clustering and builds further on the findings of [6], which is omitted from this thesis to avoid excessive overlap between the chapters. In [6] , pre-trained speaker embeddings are proposed as cluster features. The hypothesis here is that these speaker embedding networks are trained to detect the same speaker under different noise conditions. Therefore, if the same speaker dominates two microphones, the embeddings should be almost identical. The embedding based clusters are compared against classical features based on the same inputs as the embedding networks: Mel frequency cepstral coefficients (MFCC).

It then expands upon the evaluation of [6] in four different ways. Firstly, the evaluation is conducted on more realistically simulated data. This is the best available dataset, since, to the best of my knowledge, there is no applicable real world dataset that contains the evaluation scenarios. Secondly, the evaluation also tackles the hard case of where the sources are very close to each other. The dominance of the closest speaker to the microphones is much smaller in these scenarios. Thirdly, an evaluation towards more dynamic scenarios is carried out. In particular, the features are calculated over successively smaller time intervals. Lastly, a proof of concept evaluation is carried out on targeted clustering, where a target speaker's embedding is provided and the microphones close to that speaker should be clustered together with that target speaker. This could be useful for applications where you know the persons you want to talk to, and only this information is relevant. For instance, in a crowded bar where you only want to listen to your friends.

Chapter 6 then compares the embeddings based features from Chapter 5 with coherence based features [12]. Coherence based features depend on the linear re-

lation between the microphone signals. The hypothesis before trying this feature was that this would be less robust and that the source clusters would include more of the microphones from the background cluster, therefore including inferior microphones in the cluster. The reason is that microphones far away from any sources and from each other would have low coherence, and thus be unlikely to be clustered together, even though they are both mostly dominated by noise and reverberation. In contrast, the speaker embeddings do not suffer from these effects.

Another evaluation in this chapter is the bandwidth required for the clustering. The embeddings are features based on aggregated time information at each microphone itself, while the coherence is calculated on the time series itself, requiring the whole signal to be transmitted to the central processor. To mitigate bandwidth usage, the coherence based features are therefore also evaluated on encoded data. A codec reduces the bitrate of a signal but loses some information in the process. Here, the hypothesis was that this would break down the clustering method, as coherence is based on linear relations, and codec transforms the audio in a non-linear manner. It turns out that both hypnotises were in fact incorrect.

Then, **Chapter 7** exploits the (improved) clusters for speech separation with the help of a DNN. In this chapter, the importance of infusing the DNN with domain-specific knowledge becomes evident. This chapter hypothesises that a neural network that *extracts* the sound source associated with a specific cluster would outperform a neural network that takes in all microphones and needs to *separate* the sources. There are two main reasons for this hypothesis. Firstly, in distributed settings, there are a lot of microphones that pick up mostly reverberation and noise, which makes them less useful for the separation. The clustered microphones are of higher quality. Secondly, because of the ad-hoc nature of the microphones, an array agnostic neural network structure should be chosen. These typically perform some mean pooling across the different microphone features since this is a number and permutation invariant operation. However, this makes it hard for the network to distinguish the features of both speakers.

The chapter is also concerned with promoting the reference microphone, and how this could improve the extraction performance. The reference microphone should ideally be the most qualitative microphone for each cluster – the microphone where the target speaker is most dominant – and is given as a by-product of the soft clustering algorithms. Picking a good reference microphone was also crucial for some classical separation methods.

At last, **Chapter 8** proposed two additions to the separation methods of Chapter 7. The first addition is to allow for cross cluster information. In the classical cluster based separation framework, this was a critical first step in achieving good separation and suppression of the interfering speaker. However, Chapter 7 only considered exchanging information between microphones within a single cluster. By exchanging information across clusters, a decent reference of the interferer

would be present and suppressing this should be easier. Additionally, if the target speaker is silent for some period, the extraction methods might identify the interfering speaker as the dominant speaker and extract it. This would be omitted with the knowledge of the interferer.

The second addition is a method to increase the temporal resolution, without exploding the computational complexity. The DNN uses a convolutional layer to extract features from the time domain signal. Lowering the convolution's frame shift would increase the temporal resolution, but also the feature size and thus computational complexity. A strided convolution on the most computational required parts will reduce this again.

1.3 Publications

The chapters given above contain a selection of the papers I have contributed to during my PhD. Below is an overview of the all publications resulting from my PhD work. They have been published in one scientific journal and presented at various peer-reviewed international conferences.

1.3.1 Publications in international journals (listed in the Science Citation Index [1])

1. **Stijn Kindt**, Jenthe Thienpondt, Luca Becker, and Nilesh Madhu . *Robustness of ad hoc microphone clustering using speaker embeddings: evaluation under realistic and challenging scenarios.* Published in the EURASIP Journal on Audio, Speech, and Music Processing, 2023 (Correction published in 2024).

1.3.2 Publications in international conferences

2. **Stijn Kindt**, Alexander Bohlender, and Nilesh Madhu. *2D acoustic source localisation using decentralised deep neural networks on distributed microphone arrays.* Published in the proceedings of the 14th ITG Conference on Speech Communication (ITG 2021).
3. Yanjue Song, **Stijn Kindt**, and Nilesh Madhu. *Drone ego-noise cancellation for improved speech capture using deep convolutional autoencoder assisted multistage beamforming.* Published in the proceedings of the 25th International Conference on Information Fusion (FUSION 2022).

[1] The publications listed are recognized as 'A1 publications', according to the following definition used by Ghent University: A1 publications are articles listed in the Science Citation Index Expanded, the Social Science Citation Index or the Arts and Humanities Citation Index of the ISI Web of Science, restricted to contributions listed as article, review, letter, note or proceedings paper.

4. Warre Geeroms, Gianni Allebosch, **Stijn Kindt**, Loubna Kadri, Peter Veelaert, and Nilesh Madhu. *Audio-Visual Active Speaker Identification: A comparison of dense image-based features and sparse facial landmark-based features.* Published in the proceedings of Sensor Data Fusion: Trends, Solutions, Applications (SDF 2022).
5. **Stijn Kindt**, Alexander Bohlender, and Nilesh Madhu. *Improved separation of closely-spaced speakers by exploiting auxiliary direction of arrival information within a u-net architecture.* Published in the proceedings of the 18th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS 2022).
6. Jasper Maes, Siyuan Song, **Stijn Kindt**, Pieter-Jan Maes, Bruno Masiero, and Nilesh Madhu. *Investigating spherical head models to simulate binaural room impulses for training deep neural networks.* Published in the proceedings of the Audictive Conference (Audictive 2023).
7. **Stijn Kindt**, Jenthe Thienpondt, and Nilesh Madhu. *Exploiting Speaker Embeddings for Improved Microphone Clustering and Speech Separation in ad-hoc Microphone Arrays.* Published in the proceedings of 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2023).
8. Siyuan Song, **Stijn Kindt**, Jasper Maes, Alexander Bohlender, and Nilesh Madhu. *Influence of lossy speech codecs on hearing-aid, binaural sound source localisation using DNNS.* Published in the proceedings of the 10th Convention of the European Acoustics Association (Forum Acusticum 2023).
9. Luca Becker, **Stijn Kindt**, and Rainer Martin. *Fuzzy-clustering-supported Assignment of Smart-Speaker-based Microphone Arrays to Acoustic Sources in Reverberant Acoustic Environments.* Published in the proceedings of the 15th ITG Conference on Speech Communication (ITG 2023).
10. **Stijn, Kindt**, Martijn Meeldijk, and Nilesh Madhu. *Ad hoc distributed microphones clustering: A comparative analysis on using coherence and signal-specific features.* Published in the proceedings of the 15th ITG Conference on Speech Communication (ITG 2023).
11. Siyuan Song, **Stijn Kindt**, Jasper Maes, Alexander Bohlender, and Nilesh Madhu. *Comparative Study of LC3plus and Lyra codec on DNN-based Source Localisation for Hearing Aids.* Published in the proceedings of the 15th ITG Conference on Speech Communication (ITG 2023).
12. Jihyun Kim, **Stijn Kindt**, Nilesh Madhu, and Hong-Goo Kang. *Enhanced Deep Speech Separation in Clustered Ad Hoc Distributed Microphone Environments.* Accepted at the 25th Interspeech Conference (Interspeech 2024).
13. **Stijn Kindt**, Jihyun Kim, Hong-Goo Kang, and Nilesh Madhu. *Enhanced Deep Speech Separation in Clustered Ad Hoc Distributed Microphone Environments.* Accepted to the 18th International Workshop on Acoustic Signal Enhancement (IWAENC 2024).

14. Jonas Van Damme, **Stijn Kindt**, Siyuan Song, Jasper Maes, and Nilesh Madhu *Investigation on system bandwidth for DNN-based binaural sound localisation for hearing aids.* Accepted to the 18th International Workshop on Acoustic Signal Enhancement (IWAENC 2024).

References

[1] T. Suddendorf. *The gap: The science of what separates us from other animals*. Basic Books, 2013.

[2] M. E. Kelly, H. Duff, S. Kelly, J. E. McHugh Power, S. Brennan, B. A. Lawlor, and D. G. Loughrey. *The impact of social activities, social networks, social support and social relationships on the cognitive functioning of healthy older adults: a systematic review*. Systematic reviews, 6:1–18, 2017.

[3] E. C. Cherry. *Some experiments on the recognition of speech, with one and with two ears*. The Journal of the acoustical society of America, 25(5):975–979, 1953.

[4] F. H. Bess and B. W. Hornsby. *Commentary: Listening can be exhausting—Fatigue in children and adults with hearing loss*. Ear and hearing, 35(6):592–599, 2014.

[5] A. Shukla, M. Harper, E. Pedersen, A. Goman, J. J. Suen, C. Price, J. Applebaum, M. Hoyer, F. R. Lin, and N. S. Reed. *Hearing loss, loneliness, and social isolation: a systematic review*. Otolaryngology–Head and Neck Surgery, 162(5):622–633, 2020.

[6] S. Kindt, J. Thienpondt, and N. Madhu. *Exploiting Speaker Embeddings for Improved Microphone Clustering and Speech Separation in ad-hoc Microphone Arrays*. In ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1–5. IEEE, 2023.

[7] S. Kindt, J. Thienpondt, L. Becker, and N. Madhu. *Robustness of ad hoc microphone clustering using speaker embeddings: evaluation under realistic and challenging scenarios*. EURASIP Journal on Audio, Speech, and Music Processing, 2023(1):46, 2023.

[8] A. Bohlender, A. Spriet, W. Tirry, and N. Madhu. *Neural networks using full-band and subband spatial features for mask based source separation*. In 2021 29th European Signal Processing Conference (EUSIPCO), pages 346–350. IEEE, 2021.

[9] A. Bohlender, A. Spriet, W. Tirry, and N. Madhu. *Exploiting temporal context in CNN based multisource DOA estimation*. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29:1594–1608, 2021.

[10] B. Çakmak, T. Dietzen, R. Ali, P. Naylor, and T. van Waterschoot. *A distributed steered response power approach to source localization in wireless acoustic sensor networks*. In 2022 International Workshop on Acoustic Signal Enhancement (IWAENC), pages 1–5. IEEE, 2022.

[11] S. Gergen, A. Nagathil, and R. Martin. *Classification of reverberant audio signals using clustered ad hoc distributed microphones*. Signal Processing, 107:21–32, 2015.

[12] A. J. Muñoz-Montoro, P. Vera-Candeas, and M. G. Christensen. *A Coherence-based Clustering Method for Multichannel Speech Enhancement in Wireless Acoustic Sensor Networks*. In 2021 29th European Signal Processing Conference (EUSIPCO), pages 1130–1134. IEEE, 2021.

[13] I. Himawan, I. McCowan, and S. Sridharan. *Clustered blind beamforming from ad-hoc microphone arrays*. IEEE Transactions on Audio, Speech, and Language Processing, 19(4):661–676, 2010.

[14] S. Gergen, R. Martin, and N. Madhu. *Source separation by fuzzy-membership value aware beamforming and masking in ad hoc arrays*. In Speech Communication; 13th ITG-Symposium, pages 1–5. VDE, 2018.

[15] S. Gergen, R. Martin, and N. Madhu. *Source separation by feature-based clustering of microphones in ad hoc arrays*. In 2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC), pages 530–534. IEEE, 2018.

2 Background

This chapter will introduce a common frame on which this dissertation builds. Section 2.1 discusses the different array configurations (or array types), their respective signal models, and their advantages and disadvantages. In Sections 2.2 and 2.3, speaker localisation and speech separation will be discussed. Both classical methods and DNN methods are briefly discussed, as well as what effects different microphone setups have on the localisation or separation tasks. Section 2.4 discusses some of the effects of wirelessly connecting the microphones and Section 2.5 discusses the metrics used throughout this thesis.

2.1 Array Configurations

This dissertation will discuss four types of array configurations: A single microphone, a compact microphone array, and distributed individual microphones and microphone arrays.

2.1.1 Single Microphone

For a single microphone, the signal model and notation in this dissertation is the following:

$$y(t) = h(t) * s(t) + v(t)\,, \tag{2.1}$$

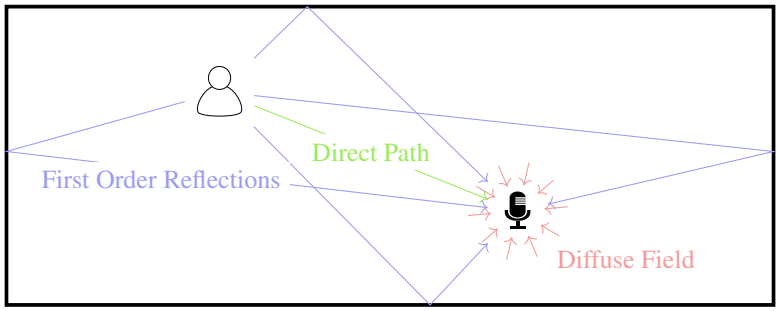


Figure 2.1: Illustration of a room with one speaker and one microphone. Some source-microphone paths, along which the speech propagates, are also plotted. More specifically, the direct path and all first order reflections (in a 2D room) are plotted, as well as a representation of the diffuse field. In the diffuse field, it seems that the sound is coming from all directions simultaneously. This happens after the signal paths have undergone enough reflections.

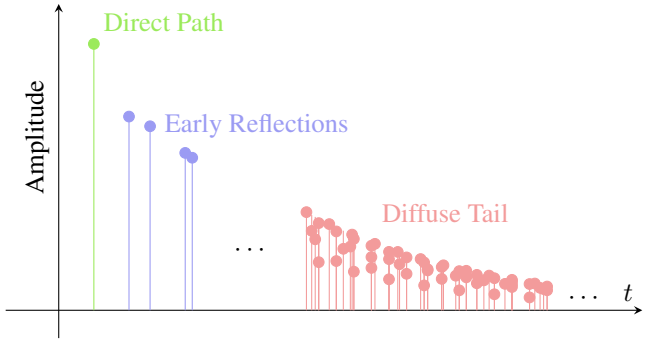


Figure 2.2: Plot of (a part of) a room impulse response (RIR). It is a series of (train of) Dirac delta peaks, each representing a speaker-microphone path. The Dirac impulses show both the time delay and amplitude of each of these reflections.

where $*$ is the convolution operator, y, h, s and v are the microphone signal, room impulse response, dry speech signal and additive noise signal respectively. t represents the continuous time instance. The room impulse response characterises the transmission path between the speaker's and microphone's locations. In Figure 2.1, the direct path and first order reflection are depicted. Additionally, the late reverberations are shown as the diffuse sound field. This translates into a room impulse response $h(t)$ which has a characteristic like that depicted in Figure 2.2. The diffuse tail has an exponential decaying characteristic [1] of reflections coming from all directions (the diffuse field).

Since the algorithms considered in this dissertation are all in the digital domain, the sampled notation is also introduced:

$$y(n) = h(n) * s(n) + v(n) = x(n) + v(n)\,, \tag{2.2}$$

where n is the discrete time index, and $*$ is now the discrete convolution. Further, the notation of the reverberant speech is shortened by $x(n) = h(n) * s(n)$ and can be decomposed as:

$$x_{j,m}(n) = x_{j,m}^{\text{dir}}(n) + x_{j,m}^{\text{rev}}(n), \tag{2.3}$$

where $x_{j,m}^{\text{dir}}$ is the direct path component and $x_{j,m}^{\text{rev}}$ represents all the reflections (early reflections as well as the diffuse tail) and thus the reverberant part of the signal.

In the speech separation scenarios, there are multiple speakers. Its signal model is:

$$y(n) = \sum_{j=1}^{J} h_j(n) * s_j(n) + v(n)\,, \tag{2.4}$$

where j is the speaker index, and J is the number of speaker in that scenario.

Further, is speech processing, it is often useful to transform the time domain signal into the frequency domain. The Fourier transform can be used for this. However, speech and many noises are non-stationary, and it is often useful to see the frequency content change over time. This is exactly the goal of the Short-Term Fourier Transforms(STFT), defined as:

$$Y(l,k) = \sum_{k'=0}^{K} g(k')y(k' - lL)e^{-j2\pi\frac{k'k}{K}} \tag{2.5}$$

$Y(l,k)$ is represents the signal $y(n)$ transformed by STFT. $g(n)$ represents a windowing function of length K. A window function selects a portion of the signal, on which the Fourier transform is carried out. Typically, it has unit height in the middle and tapers off towards the edges, where it reaches zero. This avoids hard cut-offs at the edges, which would result in distortions in the frequency content otherwise. Throughout this thesis, the Von Hann window or the square root Von Hann window is used, which has this tapering property. The exponential component in (2.5) represents the typical Fourier transform. Lastly, L represents the hop length, which indicates how many samples are skipped before computing the next Fourier transform on the next windowed signal. Typically, this is half of the window length $L = K/2$. This representation carries both spectral and temporal information.

2.1.2 Compact Microphone Array

By adding a second or more microphones near the first microphone, a compact microphone array is generated. This is depicted in Figure 2.3. The sound field is

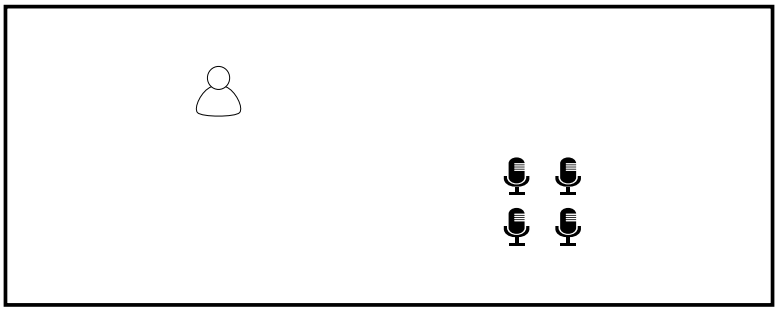


Figure 2.3: Illustration of a room with one speaker and one compact microphone array.

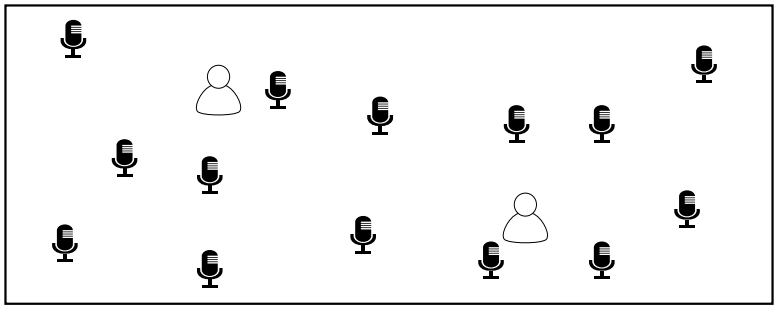


Figure 2.4: Illustration of a room with two speakers and many distributed microphones.

sampled at different locations by the microphones of the array. Next to the spectro-temporal dimensions, this gives an extra spatial dimension, which is mostly present in the time delay between microphones.

The notation (in STFT domain) for these cases is as follows:

$$Y_m(l,k) = \sum_{j=1}^{J} X_{m,j}(l,k) + V_m(l,k)\,, \tag{2.6}$$

where m is introduced as microphone index. In the compact microphone array, a total of M microphones are present.

Due to the extra spatial dimension, it is possible to determine where a sound is originating from. Also, beamforming or spatial filtering is possible. Both concepts will be explained further in Sections 2.2 and 2.3 respectively.

2.1.3 Distributed Microphones

The distributed microphone setup uses the same notation as the compact microphone setup:

$$Y_m(l,k) = \sum_{j=1}^{J} X_{m,j}(l,k) + V_m(l,k)\,. \tag{2.7}$$

The big difference is the position of the microphones. Figure 2.4 shows this setup. However, this distributed positioning does have further consequences. First of all, it is not practical that these microphones are connected via wires. Therefore, if the microphones should be jointly processed, their signals need to be wirelessly transmitted to each other or a central processor. This will require bandwidth. Lossy audio codecs can be used to lower bandwidth requirements. Wireless connectivity will be discussed in more detail in Section 2.4. Further, since the microphones are not on the same device, they also have different clocks. Although the clock rates are very close between the different devices, there are very small (typically expressed in parts per million (PPM)) differences that lead to sample rate offsets (SRO) and sample time offsets (STO). Nevertheless, after sufficient time without intervention, the samples will have drifted significantly. Unfortunately, this makes the spatial information that can be gathered from the time differences between the microphones unreliable. There are methods to counteract these SROs and STOs, for example [2, 3], but they are not perfect and a small difference can still degrade typical multi-channel processing. Even if the microphones would be perfectly synchronised, they would be subject to spatial antialiasing at relatively low frequencies due to the large distance between the microphones. This should also be kept in mind while designing the algorithms.

On the other hand, the distributed nature gives rise to another form of spatial diversity, not found in compact microphone arrays: the sound field is sampled at drastically different locations. This means that different microphones will be dominated by different sound sources. With a compact array setup, in contrast, the microphones pick up the different sound sources at similar levels. Where a compact microphone array can fail to extract a target that is far away due to nearby noise sources, chances are that a distributed array can select the right microphone that is already close to that target.

2.1.4 Distributed Microphone Arrays

The distributed microphone array is then a combination of the previous two discussed microphone setups. There are multiple arrays distributed over the room, see Figure 2.5 for an illustration. The receptive notation is:

$$Y_{a,m}(l,k) = \sum_{j=0}^{J-1} X_{a,m,j}(l,k) \quad + V_{a,m}(l,k) \tag{2.8}$$

where now a is the microphone array index. m now indicates the microphone index of array a.

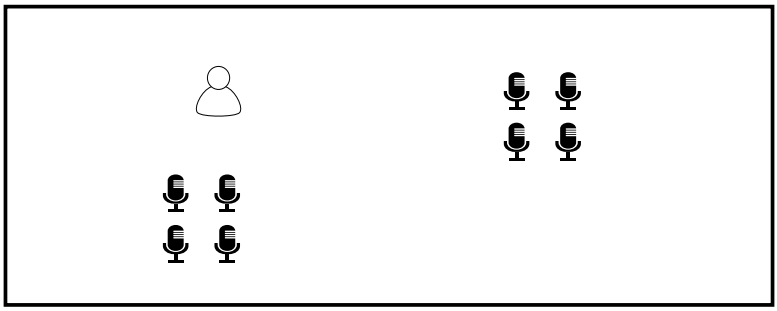


Figure 2.5: Illustration of a room with one speaker and two distributed compact microphone arrays.

This setup comes with some of the advantages of both compact microphone arrays as well as distributed individual microphones: each node a can do spectro-temporal and spatial processing, without needing to take clock asynchronicities into account, while each node separately still samples the sound field at drastically different locations (and will be dominated by different sound sources). However, the same challenges with distributed processing are still present when sharing information between microphone arrays: their clocks are not synchronised, and there is a need for wireless transmission. Chapter 4 utilises distributed microphone arrays to improve speaker localisation accuracies over single array systems.

2.1.5 Ad-Hoc Microphone Arrays

Ad-hoc microphone arrays are not so much a completely different category, as being a variant on all of the setups mentioned earlier. The main differentiating factor is that the microphones are positioned ad-hoc, and thus no prior knowledge of their location is known. In an ad-hoc compact microphone array, all the microphones are placed close to each other, but with an unknown geometry. This still provides spatial diversity, but localising sound sources is impossible. Ad-hoc compact microphone algorithms can, for example, be useful if you want to design a system that can run on multiple devices with different form factors. They can also deal with individual microphone failure.

For distributed microphone scenarios, not knowing the position of the microphones is a very common assumption. Use cases here tend to be more towards connecting all microphone carrying devices (*e.g.*phones, laptops) in an ad-hoc manner. This makes it so, besides the wireless connection, and different clocks, there is an added challenge of unknown microphone positions. Yet, their spatial diversity gives so many opportunities. Chapters 5 to 8 explores how to deal with (part of) these challenges while reaping the benefits from their diversity.

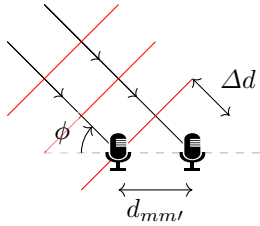


Figure 2.6: Time difference of arrival (TDOA) due to the extra distance Δd the plane wave needs to travel. This distance is dependent on the direction of arrival ϕ and the distance between the microphones $d_{mm\prime}$. The plane propagation is represented by the red lines, while the direction of propagation is along the black arrows.

2.2 Localisation

The location of a sound source can be determined by comparing microphone signals of a microphone array. More specifically, the time difference of a signal impinging on the array between the different microphones depends on the angle from which the sound is coming from. For example, in Figure 2.6, the extra distance Δd that a plane wave needs to travel from the sound source to the microphone is indicated. Dividing the distance by the speed of sound c and you get the time delay $\Delta\tau$ between the signals of the two microphones: $\Delta\tau = \frac{\Delta d}{c}$. And Δd is directly related to the DOA: $\Delta d = d_{mm\prime}cos(\phi)$, where $d_{mm\prime}$ is the distance between microphones m and $m\prime$. The plane wave assumption is valid if the microphones are far enough away from the source compared to their inter-microphone distance.

In the frequency domain, a time delay corresponds with a phase shift. Thus, phase differences across different microphones can indicate where the sound sources are located. These phase differences are a core concept of many localisation algorithms, both classical and DNN-based. This shows that these phase differences carry a lot of spatial information. They are therefore also very useful for spatially selective speech separation.

Some techniques, relevant to the thesis, will be discussed below. Sections 2.2.1 and 2.2.2 discuss some classical and DNN based localisation approaches with a single microphone array respectively. A more general overview on localisation with a single microphone array is found in [4]. Localisation with multiple microphone arrays is briefly discussed in Section 2.2.3. Lastly, Section 2.2.4 discusses weak localisation with ad hoc distributed microphones or microphone arrays.

2.2.1 Classical Methods

Some examples of classical localisation algorithms are Steered-Response Power with Phase Transform (SRP-PHAT) [5] and Generalized Cross-Correlation with Phase Transform (GCC-PHAT) [6]. GCC-PHAT aims to estimate the time delay

between the signals on the microphones. This is done in frequency domain (or at least STFT domain), as this allows for more precision compared to time domain counterparts. In time domain, the precision is limited by the sample rate. Mathematically, GCC-PHAT estimates the time varying $\widehat{\Delta\tau}$ as:

$$\widehat{\Delta\tau}(l) = \underset{\Delta\tau}{\operatorname{argmax}} \sum_{k=0}^{K} \frac{Y_m(l,k)Y^*_{m\prime}(l,k)}{|Y_m(l,k)Y^*_{m\prime}(l,k)|} e^{-j2\pi \frac{\Delta\tau k}{K}} , \tag{2.9}$$

where $\frac{1}{|Y_m(l,k)Y^*_{m\prime}(l,k)|}$ is the Phase Transform (PHAT) normalisation term. Here the time delay is estimated with one microphone pair, but if more are available, an average over all pairs can make the estimate more robust.

In SRP-PHAT, the goal is to filter out signals coming from a look direction. Then the energy from the filtered direction is calculated and the directions which have much higher energy than the others are said to contain a sound source. The beamformers can be designed in different way. Section 2.3.2 indicates a few choices. Mathematically, SRP-PHAT is expressed as:

$$\widehat{\phi}(l,k) = \underset{\phi}{\operatorname{argmax}} \frac{|\boldsymbol{W}^H(\phi,l,k)\boldsymbol{Y}(l,k)|^2}{\|\boldsymbol{Y}(l,k)\|_2^2} , \tag{2.10}$$

where $\|.\|_2$ is the ℓ_2 norm and $\boldsymbol{W}(\phi,l,k)$ are the (fixed) beamformer weights towards look direction ϕ. Here, $\frac{1}{\|\boldsymbol{Y}(l,k)\|_2^2}$ is the notation of the PHAT normalisation. Bold letters indicate the vector microphone signal notation: $\boldsymbol{Y}(k) = [Y_1(k), \cdots, Y_M(k)]^T$, where each element corresponds with one microphone signal.

2.2.2 DNN-based Methods

DNNs have also been designed to localise sound sources. An extensive overview can be found in [7]. The inputs are often the phases of the microphone signals, similar to what the classical methods use. As an output, two main options are available: classification vs regression. In the classification formulation, the space (room) is divided into different regions, and the goal of the network is to indicate in what region the source is located. The regions can either be angular sections around a microphone array [8, 9], or a grid pattern covering the whole room as in Chapter 4. The latter one is only useful if the distance can also be estimated, or in case multiple microphone arrays are collaborating to make a joint localisation. Regression is where the network outputs the location on directly. This can either be a DOA estimate or an estimate in Cartesian coordinates. [10] compared the two methods for scenarios of multiple speakers, and concluded that classification is preferable. In single source scenarios, there was less of a difference.

2.2.3 Distributed Arrays

With distributed arrays, it is possible to pin down the position of a source in Cartesian coordinates instead of only estimating their angle. This is possible by triangulation: both arrays share their respective DOA estimates, and where the estimates cross, is the position of the sound source. These DOA estimates can originate from either a classical method, or a DNN-based method. In Chapter 4 however, a different strategy is proposed, where not the DOA estimates between arrays are combined, but earlier latent features of DOA estimating DNNs. This is to improve the localisation accuracy.

2.2.4 Ad-hoc Distributed Microphone (Arrays)

While in the previously discussed localisation methods, the goal was always to localise a source. However, with ad-hoc distributed microphones, it is impossible to localise a source since the microphone positions are known, and a single microphone does not have any localisation capabilities. What is possible, however, is to blindly estimate which microphones are near the same sound source. This is a form of weak localisation. For this clustering of microphones can be used. For example, in [11–13], they extract information related to the relative spatial distance between the microphones in the form of estimating RIR of computing the inter-microphone coherence. In [14, 15], Magnitude Squared Coherence (MSC) features are extracted to extract information about the acoustic environment to cluster on. Others have used latent features on which to cluster. Examples of latent features are the Mod-MFCC features proposed in [16] and auto-encoder bottleneck features in [17]. Target outputs might look something like figure Figure 2.7, where the colours of the microphones indicate to which cluster they belong. A new cluster feature is proposed, evaluated and compared to other methods in Chapters 5 and 6 of this thesis.

With ad-hoc distributed microphone arrays, even though their position is also not known beforehand, it is possible to perform localisation. However, in this setting, both the source and array localisations need to be estimated. This can be done in an iterative manner, where the source estimates are used to localise the arrays, and in the next iteration, these new (improved) array positions are used to localise the sound sources. This is continued till both estimates have converged. One example of such an approach can be found in [18]. This method of estimating both the sources and the arrays is also called microphone array geometry calibration.

2.3 Speech Enhancement and Separation

Speech separation aims to extract all speakers present in a scenario, reducing the noise and other speakers in the process. In the following, a subset of methods,

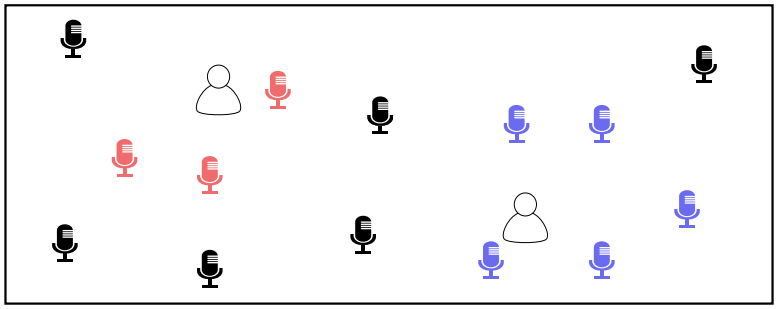


Figure 2.7: Illustration of a room with two speakers and many distributed microphones. The microphones are colour coded to indicate to which cluster they belong. The red and blue microphones each belong to a different speaker, while the black microphones do not belong to a specific speaker, and thus capture more of the background noise and reverberations.

relevant to the rest of the thesis, will be discussed. Other approaches are mentioned in the overview of [19].

2.3.1 Mask Based Enhancement and Separation

The idea of mask-based separation is to identify which Time-Frequency (TF)-bins are dominated by the target speech and which are dominated by noise.

Mask based speech separation works on the assumption of W-disjoint orthogonality [20], which says that at most one source is dominant at any TF-bin. Main reasons being that different speakers have different pitches, pronounce different phonemes and often not speak at the same time. Therefore, it is possible to estimate the speech of source j with its corresponding mask $\mathcal{M}_j(l,k)$ as:

$$\widehat{S_j}(l,k) = \mathcal{M}_j(l,k)Y(l,k) \tag{2.11}$$

The Wiener filter can be used as mask:

$$\mathcal{M}^{\text{Wiener}}(l,k) = \frac{\Phi_s(l,k)}{\Phi_y(l,k)} = \frac{\Phi_y(l,k) - \Phi_v(l,k)}{\Phi_y(l,k)}\,, \tag{2.12}$$

where Φ_s, Φ_y and Φ_v are the speech, microphone and noise Power Spectral Densities (PSDs). Φ_y can be calculated as $\Phi_y = |Y(l,k)|^2$. Φ_s and Φ_v need to be estimated.

Φ_v can for example be estimated with noise floor tracking. Some examples are minimum statistics [21], logarithmic baseline tracing [22] and minimum mean square error (MMSE)[23]. Φ_s can then be estimated by $\Phi_y(l,k) - \Phi_v(l,k)$.

However, these noise floor based separation techniques all assume that the noise is much more stationary than speech. Therefore, a noise floor tracker can

estimate the noise power, while components peaking over the noise floor are assumed speech. However, in non-stationary cases, or in cases where you want to separate two speakers, a noise floor tracker does no longer suffice.

For the single microphone case, speaker separation can be done with a Non-negative Matrix Factorisation (NMF) [24] on the STFT bins. The assumption here is that the different sources have different basis vectors, which can be decomposed. Estimating the correct reassembling vectors (masks) with NMF can then provide the clean estimate.

In the multi-microphone case, Independent Component Analyses (ICA) [25] or Independent Vector Analyses (IVA) [26] can be deployed to similarly identify which TF-bins belong together (and to the same source) and thus generate a TF-mask.

2.3.2 Beamforming

In the multichannel cases, beamformers can be deployed. They combine the signals from different microphones to form a spatial filter. Ideally, they let through all content from the direction in which the target speaker is present, while suppressing all other directions. Beamforming is defined as follows:

$$\hat{S}(l,k) = \boldsymbol{W}^H(l,k)\boldsymbol{Y}(l,k)\,, \tag{2.13}$$

where $\boldsymbol{Y}(l,k)$ is the vector notation of the different microphones: $\boldsymbol{Y}(k) = [Y_1(k), \cdots, Y_M(k)]^T$ and $\boldsymbol{W}(l,k)$ are the (adaptive) beamformer weights, that change over l. Note that fixed beamformers, independent of l, are also very useful for scenarios where there is a fixed (spatial) regions of interest. As an example, you can make a cardioid microphones by combining two microphones and a fixed beamformer.

The simplest beamformer is the delay and sum beamformer (DSB). As the name suggests, it delays the signals such that the target signal is aligned on all the microphones. Then the signals are added up, keeping the target signal while averaging out (filtering out) all the unwanted signals. In the frequency domain, the beamformer weights are defined as:

$$\boldsymbol{W}^{\text{DSB}}(l,k) = e^{j2\pi\frac{\Delta\boldsymbol{\tau}(\phi(l))k}{K}}, \tag{2.14}$$

where $\Delta\boldsymbol{\tau}(\phi(l))$ is a vector with all the time delays between the microphones and the reference microphone of the target signal. Note that delay is a function of $\phi(l)$ and can thus change its look direction ϕ over different time instances. For instance, if the speaker is moving, the beamformer can adapt accordingly.

While the DSB is effective at keeping the target direction undistorted, it has side lobes that let through signals from other directions. Other beamformers give flexibility to suppress particularly noisy directions. This can be done if spatial

target speech and noise statistics are utilised. Example statistical beamformers are the adaptive minimum variance distortionless response (MVDR) beamformer and the adaptive multi-channel Wiener filter (MWF). The beamformer weights defined as[27–30]:

$$\boldsymbol{W}^{\text{MVDR}}(l,k) = \frac{\boldsymbol{\Phi}_{vv}^{-1}(l,k)\boldsymbol{\Phi}_{ss}(l,k)\,\boldsymbol{e}_{\boldsymbol{\ell}}}{\text{Tr}(\boldsymbol{\Phi}_{vv}^{-1}(l,k)\boldsymbol{\Phi}_{ss}(l,k))}\,, \tag{2.15}$$

$$\boldsymbol{W}^{\text{MWF}}(l,k) = \boldsymbol{\Phi}_{yy}^{-1}(l,k)\boldsymbol{\Phi}_{ss}(l,k)\,\boldsymbol{e}_{\boldsymbol{\ell}}\,, \tag{2.16}$$

with $\boldsymbol{\Phi}_{ss}$ the speech spatial covariance matrix (SCM), $\boldsymbol{\Phi}_{vv}$ the noise SCM and $\boldsymbol{\Phi}_{yy}$ the microphone signal SCM. Tr(.) is the trace operator and $\boldsymbol{e}_{\boldsymbol{\ell}}$ is a one hot vector where the ℓ-th element is one and the others are zeros. This one-hot vector indicates which channel is the reference channel.

The speech and noise spatial covariance matrices are not easy to obtain, but can be estimated using the previously defined masks $\mathcal{M}(l,k)$. They act as the speech presence probability. This leads to the following weighted recursive averaging [31] on $\boldsymbol{Y}(l,k)$ for estimating the SCMs:

$$\hat{\boldsymbol{\Phi}}_{ss}(l,k) = \alpha\hat{\boldsymbol{\Phi}}_{ss}(l-1,k) + (1-\alpha)\mathcal{M}(l,k)\boldsymbol{Y}(l,k)\boldsymbol{Y}^{H}(l,k) \tag{2.17}$$

$$\hat{\boldsymbol{\Phi}}_{vv}(l,k) = \alpha\hat{\boldsymbol{\Phi}}_{vv}(l-1,k) + (1-\alpha)\mathcal{M}_{V}(l,k)\boldsymbol{Y}(l,k)\boldsymbol{Y}^{H}(l,k) \tag{2.18}$$

$$\hat{\boldsymbol{\Phi}}_{yy}(l,k) = \alpha\hat{\boldsymbol{\Phi}}_{ss}(l-1,k) + (1-\alpha)\boldsymbol{Y}(l,k)\boldsymbol{Y}^{H}(l,k)\,, \tag{2.19}$$

with α the averaging factor.

Distributed Settings

Beamformers have also been utilised in the distributed microphone setups. For example, in [32, 33], the DSB is used to separate different sources. In [13], the MVDR beamformer is used to combine the different microphones. They all have in common that the individual microphones are first clustered, and only the microphones of each cluster are used to enhance the corresponding speech signals. Clustering proved useful, as gradually including more microphones with fewer target source content did not improve the performance in [33], and including extra microphones in the beamforming decreased the performance in [13]

For distributed microphones *arrays*, the DANSE [34] algorithm is designed. The speech enhancement/separation is performed with a combined MVDR beamformer. However, instead of sending all the microphone signals to a central processor, a more bandwidth-efficient method is implemented: each array computes one signal from all its microphone signals. Only that signal is sent to another node, which combines the new signal with all its signals to a new enhanced signal and sends it to the next node and so forth. This way, all the microphone signals are still fused in the end, but no central processor is required. Sequential round robin fusion [34] and simultaneous node parameter update [35] methods are explored.

2.3.3 DNN Based Separation

DNNs can also be used to separate the different speakers. DNNs have shown superior performance compared to classical approaches, *e.g.*in [36]. DNNs can directly learn from a lot of representative data, instead of the need to design a sufficiently comprehensive model. Still within DNN design, there are many options and many venues to optimise performance. Different architectures, loss functions and input representations are being tested and compared. comprehensive overviews can be found in [19, 37, 38].

One way to utilise a DNN for speech separation is to let it estimate the TF-masks $\mathcal{M}(l, k)$ for each speech source [31]. Alternatively, a DNN can be used to clean up the noise STFT map and directly output the clean version [39]. The latter is called spectral mapping. Both methods, and a hybrid method are compared for compact microphone arrays in [40].

Similarly, DNN can also act upon the raw time domain signal directly [41], and output a cleaned up signal. Mask-based time domain DNN are also very popular [42–44]: the DNN extracts features from the time domain signal on which a mask is applied and an enhanced time domain signal is reassembled based on these masked features.

Both STFT and time domain masking are used in this thesis. Note that in the case of STFT based masking, the DNN can estimate the statistical beamformers from Section 2.3.2 to enhance the signal instead.

2.4 Wirelessly Connected Microphones

In distributed microphone setups, there is a need for a wireless transmission protocol. Audio transmission can, for example, be done over radio transmission in walkie-talkies. Another widely used method for wireless audio transmission is Bluetooth. It is, for example, used is most headsets and earbuds today. It will become even more omnipresent with the recent advances in Bluetooth low energy (BLE), which supports audio as one of its core specifications [1]. Low Complexity Communication Codec (LC3) [2] is for instance an audio codec that all BLE devices should support [45]. Audio codecs are very useful, as they lower the bandwidth required to send audio to process further.

During this thesis, the LC3 codec is the only audio codec that is used. Therefore, only this codec will be briefly discussed. The codec uses the Modified Discrete Cosine Transform (MDCT) to transform the signal into subbands and Time-Domain Aliasing Cancellation (TDAC)[46] for a perfect inverse function. Both techniques can be found in many other audio codecs as well [47].

[1] https://www.bluetooth.com/learn-about-bluetooth/feature-enhancements/le-audio/
[2] https://www.iis.fraunhofer.de/en/ff/amm/communication/lc3.html

After the MDCT, the LC3 codec performs lossy compression to lower the bitrate of the input signal, lessening the bandwidth requirements. However, this also means that perfect reconstruction is no longer possible. To hide the lossy nature of the codec to a great extent, the codec uses perceptual audio coding, which compresses the parts of the audio where humans are less likely to notice it. Overall, the quality of the speech after compression is still quite good. However, the compression does happen in a non-linear manner. This could have a devastating impact on multichannel speech processing tasks, which often rely on this coherence.

Nevertheless, in Chapter 6, it is shown that the coherence between microphone signals that went through the LC3 codec is still sufficient to determine which microphones are close to each other. This can be explained by the perceptual audio codec, which typically does only little compression on the lower frequencies since humans perceive distortions more harshly.

Besides that most audio codecs are lossy, other challenges come with using wireless systems. Firstly, the audio codecs need processing time for encoding and decoding. Additionally, they typically cannot send their signals continuously to the receiver, sending audio in packets instead. This, however, also adds to the total system latency. With wireless systems, there is also the chance that a packet gets lost and does not arrive at the receiver. Either this requires extra retransmissions or the algorithms have to deal with missing audio data. Although these last problems are real concerns for a fully functioning system, they are not yet considered during this thesis.

A property of wireless transmission that is utilised during this thesis is that these systems require synchronisation between the different devices, in order to send their packets in the right time slot. In a fully integrated audio solution, this can counteract the SRO and STO typically present in distributed microphone setups, and can keep the audio signals weakly synchronised.

2.5 Metrics

2.5.1 Speech Enhancement Metrics

In this section, the speech enhancement metrics throughout this book will be put into context. There are 8 metrics used in this book: PESQ [48], STOI [49], SIR [50], SI-SDR [51], DRR, DRINR, SNR and accuracy.

PESQ

The Perceptual Evaluation of Speech Quality (PESQ) metric is used to measure the quality of speech. It is a metric whose goal is to approximate the Mean Opinion Score (MOS) [52]. MOS is gotten by performing (expensive and time consuming) listening tests. To avoid the hassle of doing such a test for each new algorithm,

PESQ was designed. during MOS measurements, participants are asked to score an audio fragment from a scale of 1 (unacceptable) to 5 (excellent). Similarly, PESQ ranges from slightly above 1 to slightly above 4.5.

Originally, the goal of PESQ was to evaluate the perceived quality loss of audio codecs. However, it has been used by many speech enhancement and separation works as proxy for the MOS score. Nevertheless, PESQ is not ideal to measure signal improvements in very noisy and reverberant conditions. This is because, if the PESQ is already at the minimum, it is possible to subjectively improve the quality of the signal, while the PESQ score stays at the minimum score. Still, improving PESQ correlates with a better audio quality, making the metric practically useful. For example, in the box plots of Chapters 5 to 8, the tails sometimes keep touching the minimal values even after enhancement, while the means on the other hand do show improvements.

Alternative quality metrics include Perceptual Objective Listening Quality Assessment (POLQA) [53] and Deep Noise Suppression Mean Opinion Score (DNS-MOS) [54]. However, the POLQA requires a license to use. DNSMOS is a relatively recent metric and freely available to evaluate speech enhancement algorithms and is gaining in popularity. Nevertheless, throughout this thesis, the PESQ metric is still consistently used as its quality metric.

STOI

On the other hand, Short-Time Objective Intelligibility (STOI) is used to measure Intelligibility. The metric ranges from 0 to 1, where 1 is perfect ineligibility. A STOI if 0.6 and above is intelligible, although lower values will require more focus to do so. However, having a high STOI score does not necessarily mean that there is no noise any more or distortions, as long as the speech is still understandable. For example, the LC3 codec at lower bitrates can score very high STOI scores, even when the PESQ (quality) has dropped significantly.

SIR

SIR is defined as:

$$\text{SIR}_{\text{dB}} = -10 \log_{10} \frac{||\mathcal{M}_t X_t||^2}{||\mathcal{M}_t X_i||^2}, \tag{2.20}$$

where $\|.\|_2$ is the ℓ_2 norm of a vector, $\mathcal{M}$ is the speech separation mask and S_t and S_i are the speech signals of the target and interferer respectively. The SIR metric in (2.20) is expressed in dB.

SIR is particularly interesting to evaluate how well a system can suppress the interferer. This is a very handy metric for Chapters 3, 5, and 6, where the algorithms were specifically designed to suppress the interferer better. In Chapter 3 for

example, the sources are close to each other, and therefore are harder to separate. That chapter aims to improve that separation with additional location information, making SIR a perfect tool to measure if that succeeded.

SI-SDR

Signal-invariant signal-to-distortion ratio (SI-SDR), as the name suggests, measures how much the estimated signal is distorted compared to the clean reference in a scale invariant way. The distortions account for target speech suppression (removing parts of the target speech), as well as residuals from the noise and interferer that are not fully suppressed. Thus, compared to the SIR metric, SI-SDR measures more at once and is better at getting a wholistic view on the speech enhancement and separation. The scale invariant property makes it so that even if the signal is perfectly reconstructed but scales with a constant factor, the metric would show a perfect score. SI-SDR is defined as follows:

$$\text{SI-SDR}_{\text{dB}} = -10 \log_{10} \frac{||\alpha s||^2}{||\alpha s - \hat{s}||^2}, \tag{2.21}$$

where $\alpha = (\hat{s}^T \cdot s)/||s||^2$ is a scaling factor. If instead the scaling factor is put to 1, this would be the 'normal' signals to distortion ratio (SDR), which will punish constant scaling factors.

The scale-invariant property is important for certain neural network approaches. For instance, if a network aims to estimate the target waveform directly, based on the noisy input. It is common practice to normalise the DNN's input signal to improve generalisability of the network. However, this operation makes it impossible for the network to output a signal with the correct amplitude. SI-SDR circumvents this problem.

2.5.2 Other Metrics

There are more metrics used during, that do not measure the speech enhancement capabilities. For example, in Chapters 5 and 6, two novel metrics are introduced to measure how well microphones are clustered with respect to their dominant sound source: the DRR and DRINR. Next to this, SNR is used in all chapters as a metric to indicate how much noise is added to the room. lastly, accuracy is used to compare how precise the strategies of Chapter 4 manage to localise the sound sources.

DRR and DRINR

The definitions, as first described in [55], of DRR and DRINR are as follows:

$$\text{DRR} = \frac{\sum_n (x_{j,m}^{\text{dir}}(n))^2}{\sum_n (x_{j,m}^{\text{rev}}(n))^2} \quad \text{and} \tag{2.22}$$

$$\text{DRINR} = \frac{\sum_n (x_{j,m}^{\text{dir}}(n))^2}{\sum_n (y_m(n) - x_{j,m}^{\text{dir}}(n))^2}. \tag{2.23}$$

As the names suggest, it measures how much energy is in the direct component of a source-microphone pair compared to the energy of the reverberant parts (early reflections as well as diffuse tail) or the reverberant, noise and interfering energy in the scenario. The goal of the metric is to indicate how high the quality of each microphone is with respect to the target signal with index j.

In the clustering task, the aim is to have as many high quality microphones, while avoiding low quality microphones. The quality of the microphone depends on how dominant the target speaker is on each microphone, compared to other components, such as reverberation, noise and interference. Comparing histograms of the DRR and DRINR metrics between different clustering methods will give an indication which method misses some high quality microphones and includes some lower quality microphones in their clusters. Both undesirable outcomes.

SNR

Signal to noise ratio (SNR) measures the energy of the signal compared to the noise energy. Although SNR can be used as a speech enhancement metric, it is never used as such in this thesis. The main reason is that SIR and SI-SDR (or normal SDR) are better metrics in multi-source scenarios, where the target is to suppress both the noise and the interferer. The SNR only measures the energy of the noise as negative components.

However, this metric is used in this thesis to determine what the non-speech noise level is compared to the speech content of both speakers. This way the speech separation of speaker localisation can be evaluated under different noisy conditions.

Accuracy

The last metric is accuracy. This is only used to measure how accurately the localisation is. Its definition is the number of correct classifications compared to the number of incorrect classifications. This metric is limited in the range from zero (got nothing correct) to one (perfect accuracy).

References

[1] D. Howard and J. Angus. *Acoustics and psychoacoustics*. Routledge, 2013.

[2] S. Markovich-Golan, S. Gannot, and I. Cohen. *Blind Sampling Rate Offset Estimation and Compensation in Wireless Acoustic Sensor Networks with Application to Beamforming*. In International Workshop on Acoustic Signal Enhancement (IWAENC), pages 1–4, 2012.

[3] T. Gburrek, J. Schmalenstroeer, and R. Haeb-Umbach. *On Synchronization of Wireless Acoustic Sensor Networks in the Presence of Time-Varying Sampling Rate Offsets and Speaker Changes*. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 916–920, 2022. doi:10.1109/ICASSP43922.2022.9746284.

[4] D. Desai and N. Mehendale. *A review on sound source localization systems*. Archives of Computational Methods in Engineering, 29(7):4631–4642, 2022.

[5] J. H. DiBiase. *A high-accuracy, low-latency technique for talker localization in reverberant environments using microphone arrays*. Brown University, 2000.

[6] C. Knapp and G. Carter. *The generalized correlation method for estimation of time delay*. IEEE transactions on acoustics, speech, and signal processing, 24(4):320–327, 1976.

[7] P.-A. Grumiaux, S. Kitić, L. Girin, and A. Guérin. *A survey of sound source localization with deep learning methods*. The Journal of the Acoustical Society of America, 152(1):107–151, 2022.

[8] S. Chakrabarty and E. A. P. Habets. *Multi-Speaker DOA Estimation Using Deep Convolutional Networks Trained With Noise Signals*. IEEE Journal of Selected Topics in Signal Processing, 13(1):8–21, 2019. doi:10.1109/JSTSP.2019.2901664.

[9] A. Bohlender, A. Spriet, W. Tirry, and N. Madhu. *Exploiting temporal context in CNN based multisource DOA estimation*. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29:1594–1608, 2021.

[10] P. Cooreman, A. Bohlender, and N. Madhu. *CRNN-based multi-DOA estimator: Comparing classification and regression*. In Speech Communication; 15th ITG Conference, pages 156–160. VDE, 2023.

[11] S. Pasha, Y. X. Zou, and C. Ritz. *Forming ad-hoc microphone arrays through clustering of acoustic room impulse responses*. In 2015 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP), pages 84–88. IEEE, 2015.

[12] M. Dziubany, R. Machhamer, H. Laux, A. Schmeink, K.-U. Gollmer, G. Burger, and G. Dartmann. *Machine Learning Based Indoor Localization Using a Representative k-Nearest-Neighbor Classifier on a Low-Cost IoT-Hardware*. In 2018 26th European Signal Processing Conference (EUSIPCO), pages 2050–2054, 2018. doi:10.23919/EUSIPCO.2018.8553155.

[13] I. Himawan, I. McCowan, and S. Sridharan. *Clustered blind beamforming from ad-hoc microphone arrays*. IEEE Transactions on Audio, Speech, and Language Processing, 19(4):661–676, 2010.

[14] A. J. Muñoz-Montoro, P. Vera-Candeas, and M. G. Christensen. *A Coherence-based Clustering Method for Multichannel Speech Enhancement in Wireless Acoustic Sensor Networks*. In 2021 29th European Signal Processing Conference (EUSIPCO), pages 1130–1134. IEEE, 2021.

[15] Y. Zhao, J. K. Nielsen, J. Chen, and M. G. Christensen. *Model-based distributed node clustering and multi-speaker speech presence probability estimation in wireless acoustic sensor networks*. The Journal of the Acoustical Society of America, 147(6):4189–4201, 2020.

[16] S. Gergen and R. Martin. *Estimating source dominated microphone clusters in ad-hoc microphone arrays by fuzzy clustering in the feature space*. In Speech Communication; 12. ITG Symposium, pages 1–5. VDE, 2016.

[17] A. Nelus, R. Glitza, and R. Martin. *Estimation of microphone clusters in acoustic sensor networks using unsupervised federated learning*. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 761–765. IEEE, 2021.

[18] T. Gburrek, J. Schmalenstroeer, and R. Haeb-Umbach. *Geometry calibration in wireless acoustic sensor networks utilizing DoA and distance information*. EURASIP Journal on Audio, Speech, and Music Processing, 2021(1):25, 2021.

[19] J. Agrawal, M. Gupta, and H. Garg. *A review on speech separation in cocktail party environment: challenges and approaches*. Multimedia Tools and Applications, 82(20):31035–31067, 2023.

[20] O. Yilmaz and S. Rickard. *Blind separation of speech mixtures via time-frequency masking*. IEEE Transactions on signal processing, 52(7):1830–1847, 2004.

[21] R. Martin. *Noise power spectral density estimation based on optimal smoothing and minimum statistics*. IEEE Transactions on speech and audio processing, 9(5):504–512, 2001.

[22] F. Heese and P. Vary. *Noise PSD estimation by logarithmic baseline tracing*. In 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 4405–4409. IEEE, 2015.

[23] T. Gerkmann and R. C. Hendriks. *Unbiased MMSE-based noise power estimation with low complexity and low tracking delay*. IEEE Transactions on Audio, Speech, and Language Processing, 20(4):1383–1393, 2011.

[24] M. N. Schmidt and R. K. Olsson. *Single-channel speech separation using sparse non-negative matrix factorization.* In Interspeech, volume 2, pages 2–5. Citeseer, 2006.

[25] D. Mika, G. Budzik, and J. Józwik. *Single channel source separation with ICA-based time-frequency decomposition*. Sensors, 20(7):2019, 2020.

[26] A. Hiroe. *Solution of permutation problem in frequency domain ICA, using multivariate probability density functions*. In Independent Component Analysis and Blind Signal Separation: 6th International Conference, ICA 2006, Charleston, SC, USA, March 5-8, 2006. Proceedings 6, pages 601–608. Springer, 2006.

[27] A. Spriet, M. Moonen, and J. Wouters. *Spatially pre-processed speech distortion weighted multi-channel Wiener filtering for noise reduction*. Signal Processing, 84(12):2367–2387, 2004.

[28] 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, 18(2):260–276, 2010.

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

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

[31] 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, 13(4):787–799, 2019.

[32] S. Gergen, R. Martin, and N. Madhu. *Source separation by feature-based clustering of microphones in ad hoc arrays*. In 2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC), pages 530–534. IEEE, 2018.

[33] S. Gergen, R. Martin, and N. Madhu. *Source separation by fuzzy-membership value aware beamforming and masking in ad hoc arrays*. In Speech Communication; 13th ITG-Symposium, pages 1–5. VDE, 2018.

[34] A. Bertrand and M. Moonen. *Distributed adaptive node-specific signal estimation in fully connected sensor networks—Part I: Sequential node updating*. IEEE Transactions on Signal Processing, 58(10):5277–5291, 2010.

[35] A. Bertrand and M. Moonen. *Distributed adaptive node-specific signal estimation in fully connected sensor networks—Part II: Simultaneous and asynchronous node updating*. IEEE Transactions on Signal Processing, 58(10):5292–5306, 2010.

[36] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis. *Joint optimization of masks and deep recurrent neural networks for monaural source separation*. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 23(12):2136–2147, 2015.

[37] D. Wang and J. Chen. *Supervised speech separation based on deep learning: An overview*. IEEE/ACM transactions on audio, speech, and language processing, 26(10):1702–1726, 2018.

[38] P. Ochieng. *Deep neural network techniques for monaural speech enhancement and separation: state of the art analysis*. Artificial Intelligence Review, 56(Suppl 3):3651–3703, 2023.

[39] Z.-Q. Wang, P. Wang, and D. Wang. *Complex spectral mapping for single- and multi-channel speech enhancement and robust ASR*. IEEE/ACM transactions on audio, speech, and language processing, 28:1778–1787, 2020.

[40] A. Bohlender, A. Spriet, W. Tirry, and N. Madhu. *Insights into magnitude and phase estimation by masking and mapping in DNN-based multichannel speaker separation*. In 2024 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW), pages 500–504. IEEE, 2024.

[41] A. Pandey and D. Wang. *TCNN: Temporal convolutional neural network for real-time speech enhancement in the time domain*. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6875–6879. IEEE, 2019.

[42] Y. Luo and N. Mesgarani. *Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation*. IEEE/ACM transactions on audio, speech, and language processing, 27(8):1256–1266, 2019.

[43] Y. Luo, Z. Chen, and T. Yoshioka. *Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation*. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 46–50. IEEE, 2020.

[44] K. Wang, B. He, and W.-P. Zhu. *TSTNN: Two-stage transformer based neural network for speech enhancement in the time domain*. In ICASSP 2021-2021 IEEE international conference on acoustics, speech and signal processing (ICASSP), pages 7098–7102. IEEE, 2021.

[45] H. Bhalla, O. Haggai, H. Bhalla, and O. Haggai. *LC3 Codec*. Unraveling Bluetooth LE Audio: Stretching the Limits of Interoperable Wireless Audio with Bluetooth Next-Generation Low Energy Audio Standards, pages 145–159, 2021.

[46] J. Princen and A. Bradley. *Analysis/synthesis filter bank design based on time domain aliasing cancellation*. IEEE Transactions on Acoustics, Speech, and Signal Processing, 34(5):1153–1161, 1986.

[47] V. A. Raj, M. D. K. Dhas, and D. Gnanadurai. *An overview of MDCT for Time Domain Aliasing Cancellation*. In 2014 International Conference on Communication and Network Technologies, pages 203–207. IEEE, 2014.

[48] 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., volume 2, pages 749–752, 2001.

[49] 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, pages 4214–4217, 2010.

[50] E. Vincent, R. Gribonval, and C. Févotte. *Performance measurement in blind audio source separation*. IEEE transactions on audio, speech, and language processing, 14(4):1462–1469, 2006.

[51] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey. *SDR–half-baked or well done?* In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 626–630. IEEE, 2019.

[52] *ITU-T Rec. P.10/G.100 (11/2017) Vocabulary for performance, quality of service and quality of experience*. 2017. Available from: https://api.semanticscholar.org/CorpusID:210127568.

[53] J. G. Beerends, C. Schmidmer, J. Berger, M. Obermann, R. Ullmann, J. Pomy, and M. Keyhl. *Perceptual objective listening quality assessment (polqa), the third generation itu-t standard for end-to-end speech quality measurement part i—temporal alignment*. journal of the audio engineering society, 61(6):366–384, 2013.

[54] C. K. Reddy, V. Gopal, and R. Cutler. *DNSMOS P. 835: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors*. In ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing (ICASSP), pages 886–890. IEEE, 2022.

[55] S. Kindt, J. Thienpondt, and N. Madhu. *Exploiting Speaker Embeddings for Improved Microphone Clustering and Speech Separation in ad-hoc Microphone Arrays*. In ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1–5. IEEE, 2023.

3

Improved Separation of Closely-spaced Speakers by Exploiting Auxiliary Direction of Arrival Information within a U-Net Architecture

This chapter deals with a single microphone array and aims to separate different speakers with a DNN. More specifically, the influence of providing sound source location information in addition to the microphone signals is evaluated. In classical beamforming techniques, the location information is often required to extract that speaker. Therefore, it is possible that this location information also helps DNN to extract speech from these directions. Two different location based input (LBI) strategies are compared: the hand-crafted 'expected phased differences' between microphones for a source positioned at that specific direction under far-field assumptions; and the more general multi hot vector representation, where the DNN learns its own representation.

⋆⋆⋆

Stijn Kindt, Alexander Bohlender, Nilesh Madhu

Published in the 18th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS 2022).

Abstract Microphone arrays use spatial diversity for separating concurrent audio sources. Source signals from different directions of arrival (DOAs) are captured with DOA-dependent time-delays between the microphones. These can be exploited in the short-time Fourier transform domain to yield time-frequency masks that extract a target signal while suppressing unwanted components. Using deep neural networks (DNNs) for mask estimation has drastically improved separation performance. However, separation of closely spaced sources remains difficult due to their similar inter-microphone time delays. We propose using auxiliary information on source DOAs within the DNN to improve the separation. This can be encoded by the expected phase differences between the microphones. Alternatively, the DNN can learn a suitable input representation on its own when provided with a multi-hot encoding of the DOAs. Experimental results demonstrate the benefit of this information for separating closely spaced sources.

3.1 Introduction

Speaker separation is the extraction of individual speech signals from a mixture of multiple overlapping talkers and additive noise. This has several use cases, such as automatic speech recognition (ASR) and transcription, which can be used in meetings or for surveillance, tele-communication devices and hearing aids [1]. Typically, separation is done in the short-time Fourier transform (STFT) domain. Because of the spectro-temporal sparsity and approximate disjointness of speech [2] in the STFT representation, a target speaker can be extracted from the mixture by selecting the time-frequency (TF) bins which are dominated by that speaker. This corresponds to applying a mask to the STFT representation of the microphone signal, where the mask has values close to 1 when the target source is dominant at the TF bin, and 0 when an interferer or noise is dominant. As an alternative to the direct application of the masks to extract the target signal, the masks can be integrated into a spatial filtering framework, where they can control the updates of the different components of adaptive beamformers. For example, in [3, 4] the masks guide the adaptation of the spatial statistics required for minimum variance distortionless response (MVDR) beamforming or the multichannel Wiener filter (MWF). For a good source separation, therefore, estimation of robust time-frequency masks for each source is the key.

If there is only one active speaker at one time, a single microphone may be sufficient to separate speech from background noise. The structure of speech is then used to detect the target signal in the noisy mixture. However, when multiple speakers are active simultaneously, they cannot be separated based on generic speech structure alone. Then additional information is needed about the specific speaker characteristics, such as the gender of the target speaker [5] or some latent space embedding of the speaker characteristics [6, 7]. With a compact microphone

array, however, multiple overlapping speakers can be separated without the need for prior knowledge on the speaker characteristics - as long as they are not co-located in space. The extra information for the source separation then comes from the spatial diversity: the time difference of arrival (TDOA) of the signals at the different microphones is dependent on the speaker locations. This information can be exploited to define appropriate time-frequency masks for the separation.

However, the spatial diversity is limited when the sources are closely spaced. Indeed, when the sources get closer, they generate increasingly similar TDOAs. This can be detrimental for separating such closely spaced sources. Note that the closeness of sources is measured by the their angle; If the sources are far away from the microphone array, this can still lead to decent distances in Cartesian coordinates. In this work, we investigate the possible advantages of adding auxiliary information, in the form of direction of arrival (DOA) information, to a deep neural network (DNN)-based mask estimation framework. Two different techniques of embedding this information are studied. The first approach uses hand-crafted features: the expected phase difference at the microphones corresponding to DOAs where active speakers are located. The second approach lets the DNN derive a suitable representation from a multi-hot encoded vector representing active speaker DOAs.

In order to generate these features, we will assume the target locations to be perfectly known. Of course, extracting this information from the microphone signals is also challenging if the sources are closely spaced but this is outside the scope of this paper. We note, however, that additional sensors, such as a camera, can help in this regard.

In terms of DNNs, architectures based on convolutional and recurrent neural layers are efficient and have been shown to perform well for speech processing, see, e.g., [8–12]. Our baseline, therefore, is a straightforward multi-channel extension of a state-of-the-art convolutional recurrent U-net architecture for speech enhancement (CRUSE), originally proposed in [9] (and optimised in [10]) for single-microphone noise suppression.

3.2 Mask-based source separation

3.2.1 Signal model

We assume that a mixture of the target speech and interference speech is captured by an M-element microphone array in a reverberant and noisy room. Thus, each captured speech signal can be obtained by convolving the dry signal at the source location with the speaker-location dependent room impulse response (RIR). So, The mixture at microphone m in a room with J speakers is:

$$y_m(n) = \sum_{j=1}^{J} h_{m,j}(n) * s_j(n) + v_m^{\text{add}}(n)\,, \tag{3.1}$$

where $s_j(n)$ is the (dry) speech signal of source j, $h_{m,j}(n)$ is the RIR modelling the direct path ($h_{m,j}^{\text{dir}}(n)$) and reflections ($h_{m,j}^{\text{ref}}(n)$) from the location of source j to microphone m, $*$ is the convolution operator and $v_m^{\text{add}}(n)$ is the additive noise at microphone m. Combining the unwanted reverberation and additive noise together, $v_m(n) = \sum_{j=1}^{J} h_{m,j}^{\text{ref}}(n) * s_j(n) + v_m^{\text{add}}(n)$, and defining $x_{m,j}(n) = h_{m,j}^{\text{dir}}(n) * s_j(n)$, we can write (3.1) in STFT domain as:

$$Y_m(l,k) = \sum_{j=1}^{J} X_{m,j}(l,k) + V_m(l,k)\,, \tag{3.2}$$

where k is the frequency index and l the frame index of the STFT. We assume that the speakers do not move during utterances, which is indeed a valid assumption in many situations *e.g.* people sitting around a table for a meeting, or at the bar. A compact representations is obtained by stacking the microphone signals into a column vector: with $\boldsymbol{X}_j(l,k) = [X_{1,j}(l,k), \cdots, X_{M,j}(l,k)]^T$ and $\boldsymbol{V}(l,k) = [V_1(l,k), \cdots, V_M(l,k)]^T$

$$\boldsymbol{Y}(l,k) = \sum_{j=1}^{J} \boldsymbol{X}_j(l,k) + \boldsymbol{V}(l,k)\,, \tag{3.3}$$

3.2.2 Separation by time-frequency masks

The well-known properties of sparsity and disjointness of speech signals in their STFT representation [2] imply that each TF bin is typically dominated by one source. Thus, by identifying and preserving the TF bins dominated by a target speaker j and suppressing the remaining TF bins, an estimate $\widehat{X}_j(l,k)$ of the target speaker signal can be obtained. In effect, this corresponds to generating a speaker specific mask $\mathcal{M}_j(l,k)$, which has values close to 1 for TF bins (l,k) dominated by $X_j(l,k)$ and values close to 0 otherwise, and applying it to the STFT spectrum of the chosen reference microphone as:

$$\widehat{X}_j(l,k) = \mathcal{M}_j(l,k) Y_{\text{ref}}(l,k)\,. \tag{3.4}$$

The separation mask can be defined in a wide variety of ways (see, e.g. [13]). Here, without loss of generality, we choose the (bounded) spectral magnitude mask (SMM):

$$\mathcal{M}_j(l,k) = \min\left(\left(\frac{|X_{\text{ref},j}(l,k)|^2}{|Y_{\text{ref}}(l,k)|^2} \right)^{\beta}; 1 \right)\,, \tag{3.5}$$

where β is a parameter that controls the trade-off between speech distortion and suppression of the interference and noise. We set $\beta = 1$, which suppresses the interferer(s) and noise more aggressively compared to the typical choice of $\beta = 0.5$, at the cost of a slightly increased distortion of the target signal. Further, as it is easier to learn a bounded target, we clip the SMM at 1. Also, without loss of generality, we assume microphone 1 is chosen as the reference.

3.3 DNN-based mask estimation

The masks needed for the separation are typically estimated using a DNN, e. g., a fully-convolutional network such as Conv-TasNet [14], which performs an end-to-end separation, or a convolutional recurrent neural network such as [8], which operates in the STFT domain. To guarantee a high speech quality when an STFT-based masking is performed, it is particularly important that the TF mask captures the *local* structure of the target. One way to accomplish this is to process frequency subbands separately, as done in [11, 12]. A more computionally efficient solution is given by encoder-decoder architectures, where local information can be preserved by means of skip connections between encoder and decoder. In recent years, many TF mask estimation approaches of this type have been proposed, e. g., [15]. In this work, we consider the optimised convolutional recurrent U-net for speech enhancement (CRUSE) structure from [10], and extend it to the multi-channel case.

3.3.1 CRUSE for multichannel separation

The extended CRUSE architecture is depicted in Figure 3.1. The convolutional layers in the first part of the U-Net (the encoder) have a kernel size of $(2, 3)$ and a stride $(1, 2)$ along the time and frequency dimension respectively. Thus, each encoder layer successively reduces the frequency dimension by half, while the feature dimension increases as depicted.

This is repeated 5 times until we get a latent space representation. The feature and frequency dimensions are then flattened to form the feature dimension for the gated recurrent unit (GRU) [16]. To reduce the complexity of the model, the features are divided into 4 groups, which are processed by 4 GRU layers of smaller size in parallel [10]. The outputs of the GRUs are 'unflattened' into the frequency and feature dimensions. Deconvolutional layers in the decoder are then used to reverse the dimensionality reduction of convolutional layers in the encoder. Additive skip connections with a learnable scaling and bias are inserted between encoder and decoder [10]. These propagate information throughout the network, and make it easier for the network to learn via back-propagation.

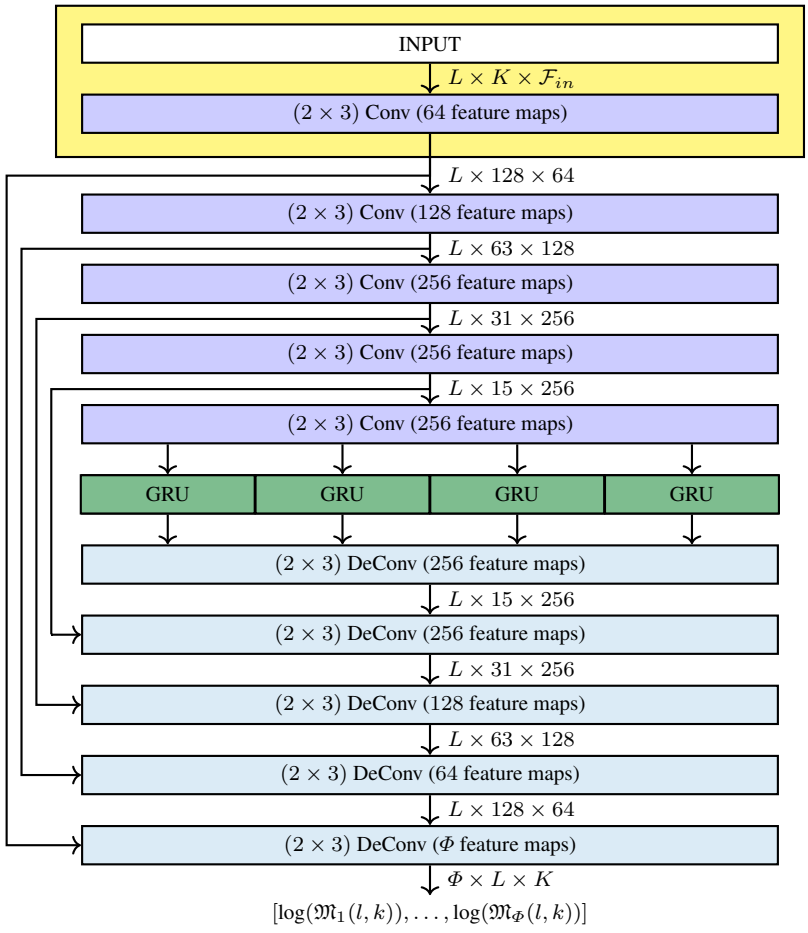


Figure 3.1: U-Net structure (assuming $K = 257$ discrete frequencies at the input). The bigger yellow box at the top will be defined by the selection of the input features. This choice of input features will also dictate the dimension ($\mathcal{F}_{in}$) of input features.

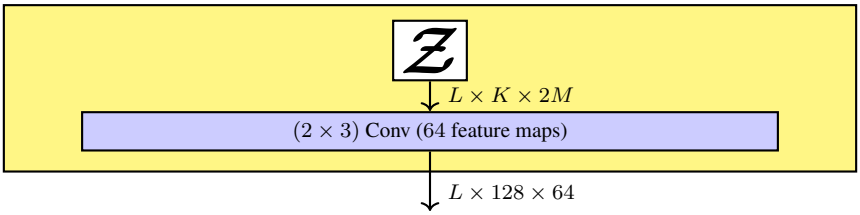


Figure 3.2: Baseline input features: $\boldsymbol{\mathcal{Z}}$ is a $L \times K \times 2M$ tensor where the third dimension is given by the vector $\boldsymbol{\mathcal{Z}}(l,k)$ defined in (3.8)*. The elements of $\boldsymbol{\mathcal{Z}}(l,k)$ consist of the real and imaginary parts of the normalised amplitudes from the microphone signals.* $\mathcal{F}_{in} = 2M$.

After each convolutional and deconvolutional layer, batch normalisation is applied. All layers, except for the output layer, use the Rectified Linear Unit (ReLU) activation function.

The yellow box in Figure 3.1 will change depending on the input features: it will either be the baseline inputs, discussed in Section 3.3.2, or one of the novel input features, incorporating auxiliary information, discussed in Section 3.4.

3.3.2 Input features

As input features for the reference (baseline) model, we straightforwardly change the single channel inputs from [10] to integrate the spatial diversity available through the microphone array. We take the real and imaginary parts of the normalised amplitude $\frac{Y_m(l,k)}{||\boldsymbol{Y}(l,k)||_2}$ at each microphone [17]:

$$\mathcal{Z}_m^R(l,k) = \Re\left\{\frac{Y_m(l,k)}{||\boldsymbol{Y}(l,k)||_2}\right\} \tag{3.6}$$

and

$$\mathcal{Z}_m^I(l,k) = \Im\left\{\frac{Y_m(l,k)}{||\boldsymbol{Y}(l,k)||_2}\right\} \tag{3.7}$$

where $\|.\|_2$ is the ℓ_2 norm of a vector.

We use the following short hand notation in Figure 3.2:

$$\boldsymbol{\mathcal{Z}}(l,k) = [\mathcal{Z}_1^R(l,k), \mathcal{Z}_1^I(l,k), \cdots, \mathcal{Z}_M^R(l,k), \mathcal{Z}_M^I(l,k)] \tag{3.8}$$

where $\boldsymbol{\mathcal{Z}}(l,k)$ is a $2M$ vector, to form the third dimension of the $L \times K \times 2M$ tensor $\boldsymbol{\mathcal{Z}}$.

Since the spatial information is essentially present in the phase, the chosen representation encodes this information well. However, compared to using the phase ($\angle Y_m(l,k)$) directly, the above representation is advantageous as it avoids the 2π phase wrapping problem.

While there is also some spatial information, like room reverberation, contained in the amplitude, normalising the amplitude across the microphones delivers (in our experience) a better generalisation to scale, speakers and also to different signal types. With this set of features, we obtain an input dimension of $\mathcal{F}_{in} = 2M$ for each TF bin.

3.3.3 Network output

For the output, we adopt the approach of [12]. The potential target locations are divided into Φ different angular sections, each corresponding to one DOA class. For each section ϕ, the network generates a mask $\mathfrak{M}_\phi(l,k)$ that can be used to extract a speaker from that direction. The correct mask for any speaker is then selected based on their (known or estimated) location: $\mathcal{M}_j(l,k) = \mathfrak{M}_\phi(l,k)$ if source j is located in angular section ϕ at time frame l (later written as $\mathbb{1}(\phi_j(l) = \phi)$). This mask is then applied as in (3.4).

The advantage of the chosen output representation, where different outputs correspond to different directions, is the implicit resolution of permutation. Thus, additional measures to resolve the permutation problem, *e.g.* permutation invariant training (PIT) [18], are not needed.

The outputs of the DNN are set to estimate the log-masks $\log(\mathcal{M}_j(l,k))$. In this manner, the dynamic range of the mask values is better utilised and a more accurate estimation of lower values is obtained. However, the log-masks have no lower bound, which is undesirable for a training target. Thus, a mingain g_{min} is imposed to limit the suppression. This is achieved by setting the output activation function to be a clipped linear function between g_{min} and 1. Another benefit of the mingain is that it can also reduce artifacts such as musical tones.

During training, we consequently minimise the mean squared error (MSE) loss between the estimated log-mask $\log \widehat{\mathcal{M}}_j(l,k)$ and the desired log-mask $\log \mathcal{M}_j(l,k)$ over all active sources, as in [12]:

$$\mathcal{L} = \sum_{l,k,j} \left(\log \widehat{\mathcal{M}}_j(l,k) - \log \mathcal{M}_j(l,k) \right)^2 . \tag{3.9}$$

Masks for directions without active speakers are treated as *don't cares* and do not contribute to the loss function.

3.4 Incorporation of auxiliary DOA information

We will show in the evaluation in Section 3.5 that this baseline system is good in separating multiple sources in general. In contrast however, the separation of closely spaces sources leaves some room for improvement. To improve upon these situations, we propose to add extra DOA information to the network. With this

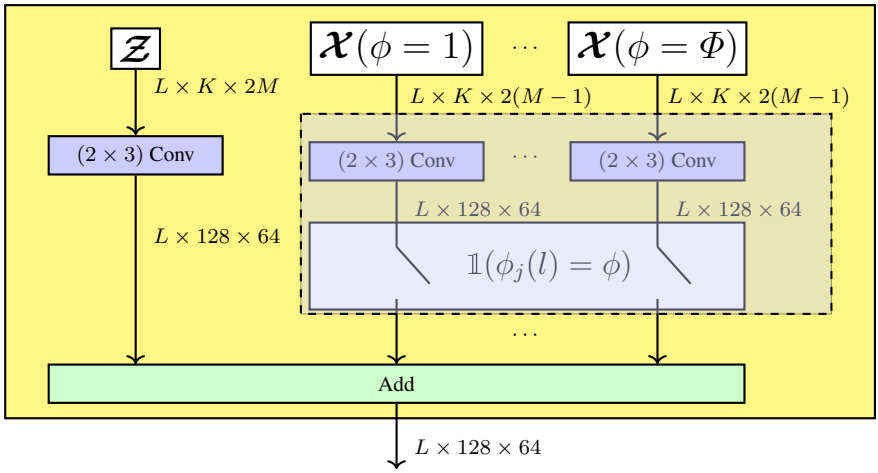


Figure 3.3: Baseline features with auxiliary expected phase difference *features* $\boldsymbol{\mathcal{X}}(\phi)$*, as defined in* (3.13)*. Additionally, the weights of the first convolutional layer are dependent on the target DOA:* $\mathbb{1}(\phi_j(l) = \phi)$*, indicated in the dotted box.* $\mathcal{F}_{in} = 2M + 2J(M-1)$.

extra information, the network should be able to extract more useful information from the very first network layer. Additionally, there should be less confusion between closely spaced sources, since the network already knows that these sources exist and are in close proximity.

We present the two different options: the first one is the use of hand crafted features, while the second one uses a multi-hot encoding of the DOA, allowing the network to learn its own representation.

3.4.1 Expected phase differences

A representation of the DOAs that permits their inclusion within the input to the neural network is given by the corresponding *expected phase difference* between the microphones of the array, as used in *e.g.* [19]. To avoid unneeded redundancy, we only take the expected phase difference between the reference (first) and mth microphone:

$$2\pi f_k \Delta\tau_m(\phi) = 2\pi f_k \tau_m(\phi) - 2\pi f_k \tau_1(\phi)\,, \tag{3.10}$$

where f_k is the central frequency at the frequency bin k and $\Delta\tau_m(\phi)$ is the time delay at microphone m, of a plane wave originating from the direction corresponding to DOA index ϕ, measured with respect to the array reference. Mathematically, $\Delta\tau_m(\phi) = [r_m^x, r_m^y][cos(\phi), sin(\phi)]^T/c$, with r_m^x and r_m^y the x- and y-coordinate of the mth channel with respect to the microphone reference microphone, and c the speed of sound.

To match the real and imaginary inputs at (3.6) and (3.7), we take the cosine

and sine of the phases at (3.10) respectively:

$$\mathcal{X}_m^C(\phi, k) = \cos\left(2\pi f_k \Delta\tau_m(\phi)\right) , \tag{3.11}$$

$$\mathcal{X}_m^S(\phi, k) = \sin\left(2\pi f_k \Delta\tau_m(\phi)\right) . \tag{3.12}$$

We make Φ tensors $\boldsymbol{\mathcal{X}}(\phi)$ of size $L \times K \times 2(M-1)$, where the third dimension is given by the $2(M-1)$ vector:

$$\boldsymbol{\mathcal{X}}(\phi, k) = \left[\mathcal{X}_2^C(\phi, k), \mathcal{X}_2^S(\phi, k), \cdots, \mathcal{X}_M^C(\phi, k), \mathcal{X}_M^S(\phi, k)\right] . \tag{3.13}$$

The same elements are repeated over the frame dimension L so the input size corresponds to $\boldsymbol{\mathcal{Z}}$. Note: as the expected phase difference for the reference microphone is always 0, it conveys no additional information and is thus not included.

This manner of including auxiliary DOA information is depicted in Figure 3.3. In order to give the network maximum flexibility, the weights of the first convolutional layer (for each $\boldsymbol{\mathcal{X}}(\phi)$) is dependent on the DOA and the corresponding expected phase differences. This means that we have Φ different sets of weights. Further, only the inputs corresponding to an (estimated) active target speaker at time frame l are passed though: $\mathbb{1}(\phi_j(l) = \phi)$. Others are multiplied with zeros as to have no influence on the mask estimation. When all J speakers are active, this yields an additional $2J(M-1)$ features per TF bin: $\mathcal{F}_{in} = 2M + 2J(M-1)$.

3.4.2 Multi-hot encoding

Alternatively, we can let the network determine a suitable representation on its own via a multi-hot input vector: we supply a Φ sized input vector for each time frame which indicates in what angular sector a speaker is active. A 1 is assigned to the ϕth element when a source is active at the location with index ϕ. This is then used as input for a fully connected (FC) layer with PK output features. The same FC layer is reused for all time frames. The encoding thus has no temporal context. We concatenate the newly generated features with the input of the baseline method by setting the convolutional layer to have an output size of PK. Here P is the number of additional input features for each TF bin. This is depicted in Figure 3.4

We also tried to increase the representation power to possibly better exploit information about which combinations of sources are concurrently active. This was done by adding additional layers between the multi-hot input and the stacking operation in Figure 3.4. However, empirically, it was found to not improve the method.

The multi-hot generated features contribute an additional P features, thus $\mathcal{F}_{in} = 2M + P$. We choose $P = 2JM$, which yields a similar number of features as for the expected phase differences.

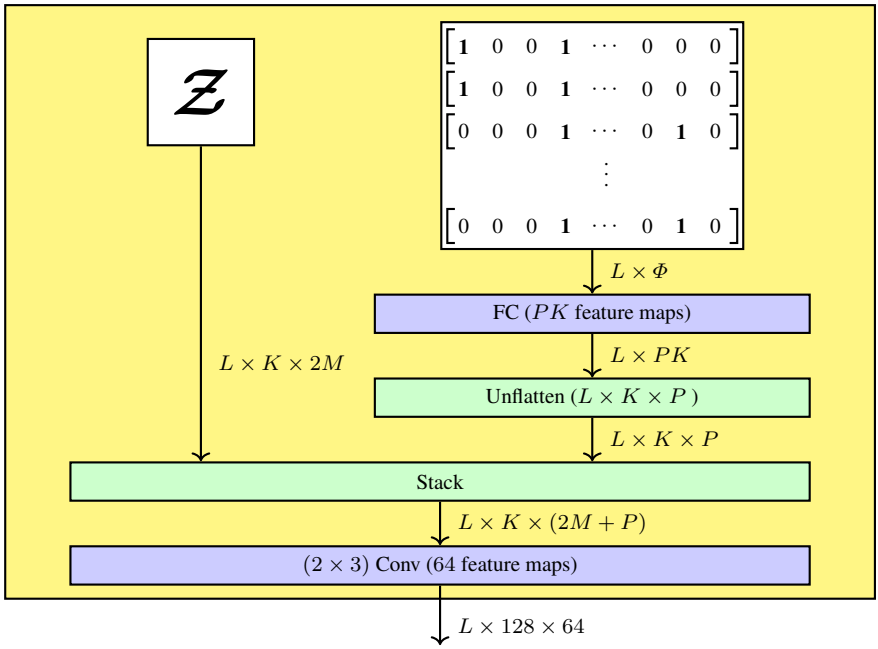


Figure 3.4: Baseline features with auxiliary features obtained from multi-hot encoding inputs. $\mathcal{F}_{in} = 2M + P$.

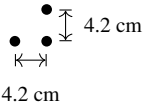


Figure 3.5: The 3-microphone array used for the experiments.

3.5 Experiments

For the experiments, we use a planar array of $M = 3$ microphones. These are placed in an isosceles right-angled triangle configuration, where the lengths of the catheti are $4.2\,cm$, as depicted in Figure 3.5. Since we use this specific array geometry, we need to generate our own train and evaluation data.

The sampling frequency is $16\,kHz$. An STFT frame length of 512 samples (with 50% overlap between frames) is chosen, resulting in $K = 257$ frequency bins (positive frequency spectrum). The g_{min} is set to $-40\,dB$.

3.5.1 Training

For training, we used the TIMIT [20] and PTDB-TUG speech datasets [21]. During the training, different scenarios are simulated where either one or two sources are

concurrently active, similar to [22]. Thus, we set maximum number of active sources J to 2. The location of each speaker is constant until the source becomes inactive (silent). When the source becomes active again, a new location is randomly assigned to the source. Source activity and inactivity are modelled as two states of a Markov chain, where a transition between the two states occurs once every $1.5\,s$ on average. See [22] for more details on the training setup.

The source signals are convolved with RIRs simulated using [23]. There are 10 different rooms with reverberation times ranging from $RT_{60} = 0.2\,s$ to $0.8\,s$. Further, for simplicity, we consider that the speakers are present only in the $180°$ angular space around the front of the array. We divide this region into angular sections of $5°$ width, resulting in $\Phi = 37$ different sections and, consequently, 37 different masks at the output. Different sets of RIRs are produced for training and validation. For the additive noise, we simulate temporally uncorrelated diffuse noise, as described by [24], with input SNRs ranging from $0\,dB$ to $30\,dB$. We stress that the network is not specifically trained to separate only closely spaced sources, since the locations are chosen arbitrarily, and there are also cases where only one speaker is active. Thereby, we can ensure that an improved separation of closely spaced sources *does not come at the cost of a reduced usefulness of the system in other scenarios*.

3.5.2 Evaluation

The test RIRs for the evaluation are simulated via a different generator: pyroomacoustics [25]. We generated 327 random scenarios. Each scenario has a random room dimension between $\{4, 4, 2\}$ and $\{8, 8, 4\}$ m. Since the focus of this work is on source separation, we only consider cases where two speakers are active. The locations of both speakers are fixed during one simulation. We make sure to simulate approximately one third of the cases where the sources are closely spaced (≤ 20 degrees apart), in order to have a representative sample size. Each source signal consists of 5 speech utterances. The utterances are taken from the TSP speech database [26]. For each room, 6 different input SNRs are generated between $-5\,dB$ and $20\,dB$, where the noise is again temporally uncorrelated and spatially diffuse.

Evaluation metrics are computed for two sets of scenarios: a set where all scenarios are included, and a set consisting only of cases where the sources are separated by 20 degrees or less. This second set consists of 114 scenarios.

3.5.3 Metrics

We consider three metrics: the first is the source-to-interference ratio (SIR), as defined by [27]. This is an important metric for source separation since it indicates how much the interferer is suppressed relative to the target.

The other metrics focus on perceptual quality (PESQ: perceptual evaluation of speech quality [28]) and intelligibility (STOI: short-time objective intelligibility [29]). These metrics offer important, complementary information on the separation performance, since the SIR alone can be misleading: a decent SIR can be achieved by suppressing all of the interfering source, while only keeping a small portion of the target speech. This would however lead to unintelligible, poor quality speech.

3.5.4 Results and discussion

In Figure 3.6, the ΔSIR metric is plotted, *i.e.*, the gain with respect to the input signal. The baseline system, without extra DOA information, yields a slightly better performance for almost every input SNR when the results for all spacings are averaged (Figure 3.6a) . However, for all three variants, the ΔSIR is very high, so that the minor difference is insignificant. In general, we can conclude that the extra DOA information does not have an influence on the performance of the CRUSE architecture. The network can infer the spatial information on its own.

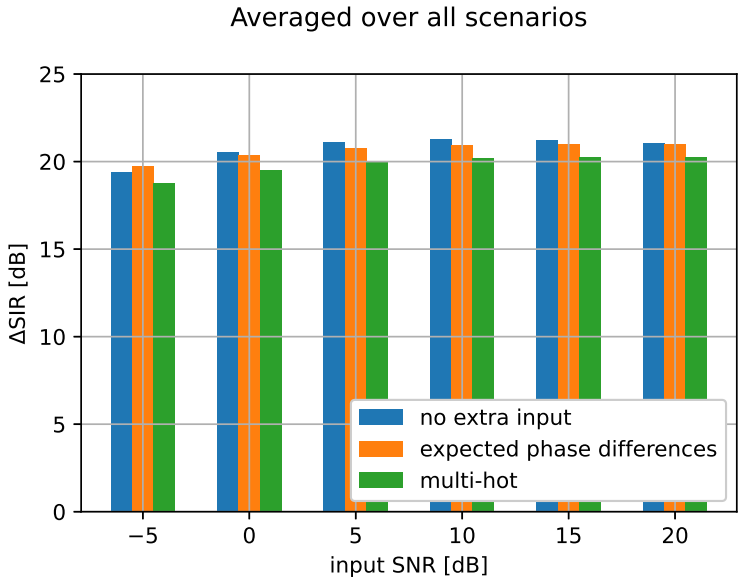
In contrast, the advantage of the extra DOA information is clearly visible when only closely spaced sources are considered (Figure 3.6b). In these scenarios, incorporating auxiliary information yields a consistent gain of 2 to $2.5\,dB$ over all input SNRs. However, the ΔSIR is less than when the sources are farther apart. This is not surprising because of the difficulty of separating sources with the considered compact 3-microphone array when their DOAs are similar.

Comparing the hand crafted expected phase difference features to the multi-hot encoding, the multi-hot encoding comes out on top for almost every case. This leads us to conclude that the network can learn a better representation than the expected phase differences to encode the DOA information. Either way, we would expect the multi-hot encoding to perform at least as well, since it could generate a representation equal to the expected phase differences.

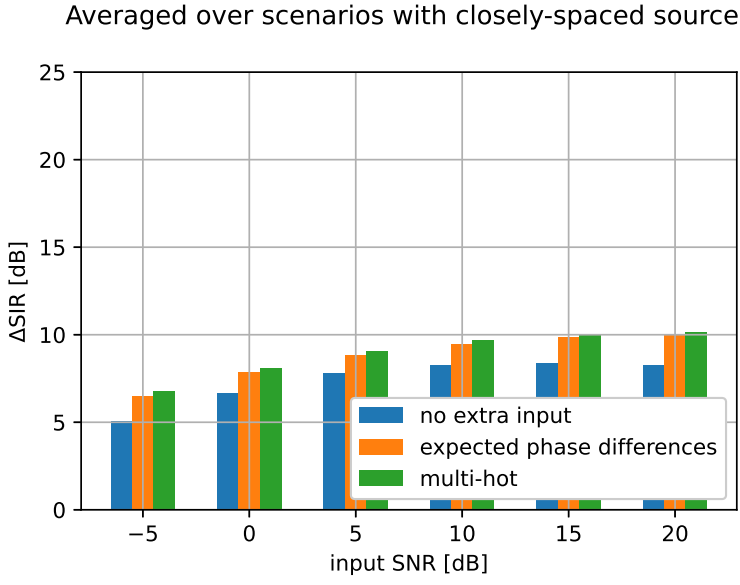
The improved interferer suppression obtained by incorporating the auxiliary information is evident when listening to examples with a spacing of 20 degrees between the two speakers. Some samples can be found at https://aspire.ugent.be/demos/AVSS2022SK/.

The STOI and PESQ graphs from Figure 3.7 and Figure 3.8 validate these informal perceptual observations, and are largely in line with what we observed in the SIR graphs: averaging over all inter source distances does not show a *significant* benefit of the additional DOA input features (even though the PESQ and STOI metrics favour these systems), but when looking at the performance for closely spaced sources only, the benefit becomes clear.

There is one outlier: the PESQ score for closely spaced sources at $-5\,dB$. Here, the original input features do seem the have an edge. However, this is not in

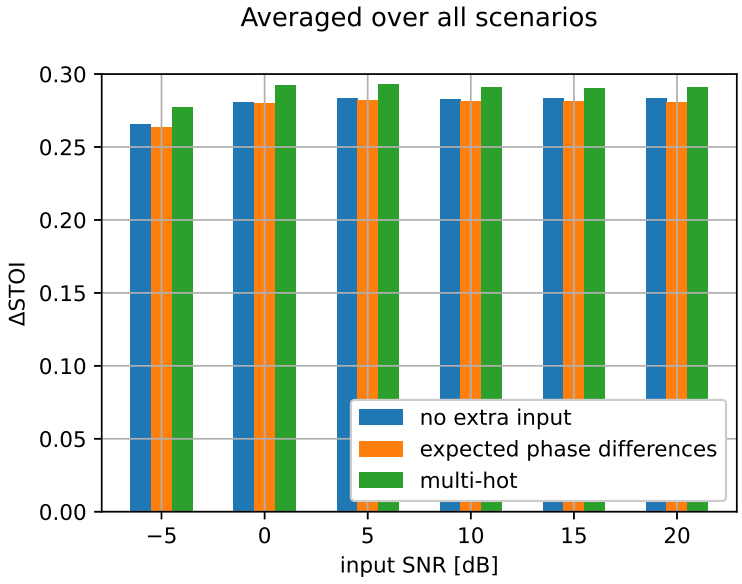


(a)

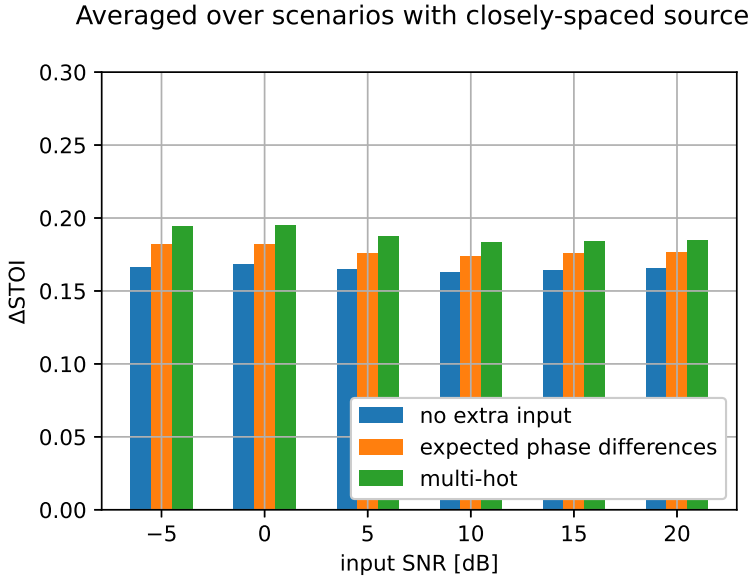


(b)

Figure 3.6: The ΔSIR metrics (as a function of different input SNRs) for all simulated cases on the top, and for the subset where sources are separated by only 20 degrees or less on the bottom. The absolute values of the metrics are added in Figure B.1 in Appendix B.

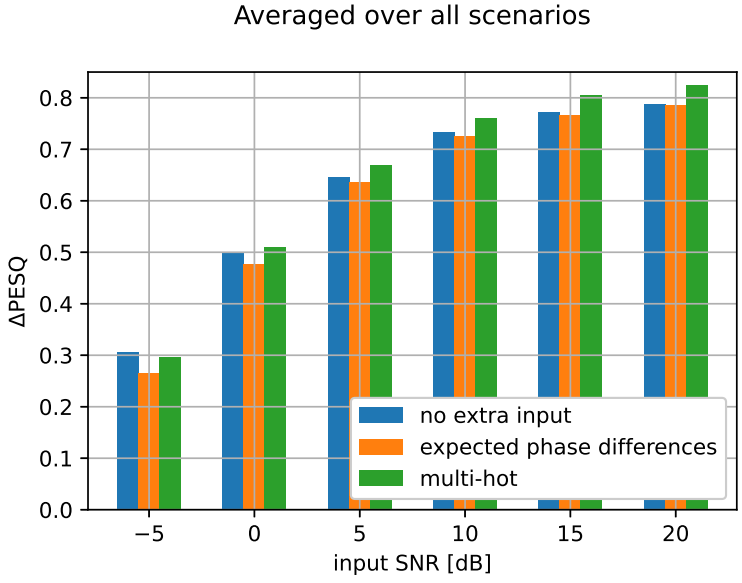


(a)

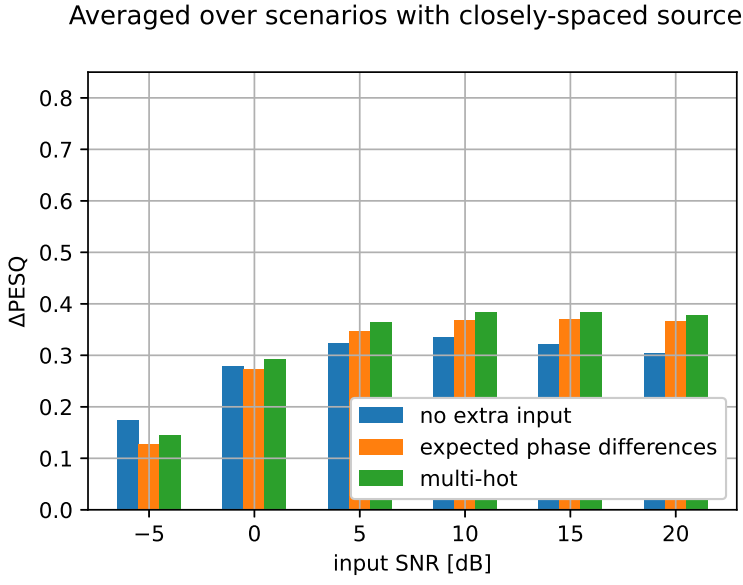


(b)

Figure 3.7: The ΔSTOI metrics for all simulated cases (upper) and the subset with only closely spaced sources (lower). The absolute values of the metrics are added in Figure B.2 in Appendix B.



(a)



(b)

Figure 3.8: The ΔPESQ metrics for all simulated cases (upper) and the subset with only closely spaced sources (lower). The absolute values of the metrics are added in Figure B.3 in Appendix B.

line with our observations when listening to the examples ourselves. This is likely because PESQ is less reliable at low input SNRs.

3.6 Conclusions

We incorporated additional DOA information at the input of a recurrent convolutional U-Net to improve closely spaced source separation with a compact microphone array.

Two representations of DOA information were considered: expected phase differences and multi-hot encoding. For sources that are farther apart, the additional inputs did not have significant impact. This shows that, generally, the network can separate sources effectively without requiring knowledge on the exact target locations.

In situations where the sources are closely spaced, on the other hand, both proposed methods were found to improve the separation. Of the two, the multi-hot encoder slightly outperformed the handcrafted expected phase difference features, indicating that the network is able to generate a superior representation.

For this work, we assumed the DOAs to be known. Future work will investigate the influence of DOA errors, resulting from estimation of the DOAs.

Acknowledgment

his work is supported by the Research Foundation - Flanders (FWO) under grant numbers G081420N and 11G0721N and imec.ICON: BLE2AV (support from VLAIO). Partners: imec, Televic, Cochlear, and Qorvo.

References

[1] A. Bertrand. *Applications and trends in wireless acoustic sensor networks: A signal processing perspective*. In 2011 18th IEEE symposium on communications and vehicular technology in the Benelux (SCVT), pages 1–6. IEEE, 2011.

[2] S. Rickard and O. Yilmaz. *On the approximate W-disjoint orthogonality of speech*. In 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing, volume 1, pages I–529. IEEE, 2002.

[3] N. Madhu and R. Martin. *A Versatile Framework for Speaker Separation Using a Model-Based Speaker Localization Approach*. IEEE Transactions on Audio, Speech, and Language Processing, 19(7):1900–1912, 2011. doi:10.1109/TASL.2010.2102754.

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

[5] D. Ditter and T. Gerkmann. *Influence of Speaker-Specific Parameters on Speech Separation Systems.* In INTERSPEECH, pages 4584–4588, 2019.

[6] M. Ge, C. Xu, L. Wang, E. S. Chng, J. Dang, and H. Li. *SpEx+: A Complete Time Domain Speaker Extraction Network*. Proc. Interspeech 2020, pages 1406–1410, 2020.

[7] R. Sinha, M. Tammen, C. Rollwage, and S. Doclo. *Speaker-conditioned Target Speaker Extraction Based on Customized LSTM Cells*. In Speech Communication; 14th ITG Conference, pages 1–5. VDE, 2021.

[8] S. Chakrabarty and E. A. P. Habets. *Time–Frequency Masking Based Online Multi-Channel Speech Enhancement With Convolutional Recurrent Neural Networks*. IEEE Journal of Selected Topics in Signal Processing, 13(4):787–799, 2019.

[9] K. Tan and D. Wang. *A convolutional recurrent neural network for real-time speech enhancement.* In Interspeech, volume 2018, pages 3229–3233, 2018.

[10] S. Braun, H. Gamper, C. K. Reddy, and I. Tashev. *Towards efficient models for real-time deep noise suppression*. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 656–660. IEEE, 2021.

[11] X. Hao, X. Su, R. Horaud, and X. Li. *Fullsubnet: A Full-Band and Sub-Band Fusion Model for Real-Time Single-Channel Speech Enhancement*. In Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6633–6637, 2021.

[12] A. Bohlender, A. Spriet, W. Tirry, and N. Madhu. *Neural networks using full-band and subband spatial features for mask based source separation*. In 2021 29th European Signal Processing Conference (EUSIPCO), pages 346–350. IEEE, 2021.

[13] D. Wang and J. Chen. *Supervised speech separation based on deep learning: An overview*. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 26(10):1702–1726, 2018.

[14] Y. Luo and N. Mesgarani. *Conv-TasNet: Surpassing Ideal Time–Frequency Magnitude Masking for Speech Separation*. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 27(8):1256–1266, 2019. doi:10.1109/TASLP.2019.2915167.

[15] S. E. Chazan, H. Hammer, G. Hazan, J. Goldberger, and S. Gannot. *Multi-Microphone Speaker Separation based on Deep DOA Estimation*. In Proc. 27th European Signal Processing Conference (EUSIPCO), pages 1–5, 2019. doi:10.23919/EUSIPCO.2019.8903121.

[16] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio. *On the Properties of Neural Machine Translation: Encoder–Decoder Approaches*. In Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, pages 103–111, 2014.

[17] Y. Yu, W. Wang, and P. Han. *Localization based stereo speech source separation using probabilistic time-frequency masking and deep neural networks*. EURASIP Journal on Audio, Speech, and Music Processing, 2016(1):1–18, 2016.

[18] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen. *Permutation invariant training of deep models for speaker-independent multi-talker speech separation*. In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 241–245. IEEE, 2017.

[19] P. Pertilä and J. Nikunen. *Distant speech separation using predicted time–frequency masks from spatial features*. Speech communication, 68:97–106, 2015.

[20] J. S. Garofolo. *Timit acoustic phonetic continuous speech corpus*. Linguistic Data Consortium, 1993, 1993.

[21] G. Pirker, M. Wohlmayr, S. Petrik, and F. Pernkopf. *A pitch tracking corpus with evaluation on multipitch tracking scenario*. In Twelfth Annual Conference of the International Speech Communication Association, 2011.

[22] A. Bohlender, A. Spriet, W. Tirry, and N. Madhu. *Exploiting temporal context in CNN based multisource DOA estimation*. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29:1594–1608, 2021.

[23] E. A. Habets. *Room impulse response generator*. Technische Universiteit Eindhoven, Tech. Rep, 2(2.4):1, 2006.

[24] E. A. Habets and S. Gannot. *Generating sensor signals in isotropic noise fields*. The Journal of the Acoustical Society of America, 122(6):3464–3470, 2007.

[25] R. Scheibler, E. Bezzam, and I. Dokmanić. *Pyroomacoustics: A python package for audio room simulation and array processing algorithms*. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 351–355. IEEE, 2018.

[26] P. Kabal. *TSP speech database*. McGill University, Database Version, 1(0):09–02, 2002.

[27] E. Vincent, R. Gribonval, and C. Févotte. *Performance measurement in blind audio source separation*. IEEE transactions on audio, speech, and language processing, 14(4):1462–1469, 2006.

[28] 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., volume 2, pages 749–752, 2001.

[29] 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, pages 4214–4217, 2010.

4

2D Acoustic Source Localisation Using Decentralised Deep Neural Networks on Distributed Microphone Arrays

This chapter sets a step towards scenarios with ad-hoc *distributed microphone arrays. However, the locations of the microphone arrays are assumed known with respect to each other, making the methods not yet applicable to truly ad-hoc scenarios. Nevertheless, the method is still flexible with respect to the different room sizes and exact positions of the microphone arrays within that room. The task is to localise the sound sources, where the co-operation between the microphone arrays can improve the localisation accuracy. Certainly, the distributed setup could be of particular interest in cases where the sources are closely spaced. Then the improved localisation can be fed back into the location informed separation networks of Chapter 3. In the design of the separation network, special care is taken that the proposed methods are robust against sample rate offsets and do not suffer from spatial aliasing.*

⋆ ⋆ ⋆

Stijn Kindt, Alexander Bohlender, Nilesh Madhu

Published in 14th ITG Conference on Speech Communication (ITG 2021).

Abstract This paper takes a previously proposed convolutional recurrent deep neural network (DNN) approach to direction of arrival (DoA) estimation and extends this to perform 2D localisation using distributed microphone arrays. Triangulation on the individual DoAs from each array is the most straightforward extension of the original DNN. This paper proposes to allow more co-operation between the individual microphone arrays by sharing part of their neural network, in order to achieve a higher localisation accuracy. Two strategies will be discussed: one where the shared network has narrowband information, and one where only broadband information is shared. Robustness against slight clock offsets between different arrays is ensured by only sharing information at deeper layers in the DNN. The position and configuration of the microphone arrays are assumed known, in order to train the network. Simulations will show that combining information between neural network layers has a significant improvement over the triangulation approach.

4.1 Introduction

Wireless acoustic sensor networks (WASNs) is a field that is gaining a lot of attraction recently. Multiple microphones or microphone arrays are distributed around a room, in order to have a bigger coverage. This can aid in a variety of applications [1], like speech enhancement [2], beamforming [3], environment monitoring [4], hands-free communication and speaker localisation [5].

When only a single microphone array is present, it is very possible to estimate the direction of arrival (DoA) of the signal coming from a speaker as long as the direct path from the source to the array is dominant. In fact, a lot of research has already been done on this topic and an overview of the classical methods for DoA estimation may be found in [6, Ch. 6, P. 135-170]. Neural networks led to recent advances in the field [7–10]. However, DoA estimation does not provide the distance of the source to the microphone array, and thus does not localise the source in 2D space. Techniques based on the use of data-based approaches such as deep neural networks (DNN), have tried to address this problem [11, 12]. However, distance information can not easily be inferred from the phase or amplitude. Other measures like the coherent-to-diffuse power ratio need to be used [13].

This is where a WASN, composed of distributed microphone arrays can help. Such configurations capture the source signal from widely different positions in the room, thereby permitting a better localisation estimate in the 2D space when the data from these arrays is combined.

One should, however, also be aware of the extra challenges that come with localisation using the distributed nodes in a WASN. Typically the different nodes (microphones or microphone arrays) are only weakly synchronised, meaning that the clock signals can differ fractions of a sample, up to a few samples. Comparing signals with unsynchronised clocks will lead to incorrect localisation [14–16].

WASNs generally also have limitations on the bandwidth of each node.

In ad-hoc WASNs, an additional challenge occurs: the positions of the sensor nodes are also unknown. In that case, both the source and the array positions need to be estimated, mostly leading to an iterative approach [17–19], making the system a lot more complex. This will not be further discussed here. The focus of this paper is on the use of WASNs with a known configuration (i.e., the location and configuration of each node is known and remains static).

Triangulation is an example of source localisation in 2D space using a WASN with multiple microphone arrays: each array estimates a DoA, and these estimates can be aggregated under the appropriate geometric constraints to provide an intersection point where the source is located [20–22].

Triangulation approaches easily fit the synchronisation and bandwidth limitations of WASNs: Triangulation is inherently robust against asynchronous clocks on different nodes, since the DoA computations can be done independently on each node. furthermore, by only sending the DoA to the central unit, the throughput is very low. However, not sharing more information between nodes limits the potential for higher localisation accuracy. Also, triangulation fails when the DoAs do not intersect, which can occur e.g., when the cumulative errors in the DoA estimates are large and in opposing directions.

This paper proposes three architectures that extend a deep convolutional recurrent neural network DoA approach [10]. The model will be expanded from using one single microphone array to being a WASN with two microphone arrays. A first, rather straightforward, approach triangulates two DoAs from the two different arrays. This will serve as the reference method. In order to improve the localisation accuracy, we also propose two architectures where the DNN structure is suitably modified in order to mix information between the different nodes at different depths in the DNN. These architectures will be referred to as co-operative localisation architectures (CLAs).

The architectures are designed such that they do not directly combine the information at the microphone level between the two arrays. Instead, the information exchange occurs at a deeper layer. Thereby they are robust to clock offsets, even without explicitly training for these offsets. In order to share the information, the different nodes send it to a central processing unit. While such information sharing requires a larger bandwidth, it may be an acceptable tradeoff against the higher localisation accuracy obtained.

Further, the output of the two proposed architectures will be changed. Instead of yielding two DoAs for subsequent triangulation, the network will directly output a 2D estimate of the source position. This solves the problem where triangulation does not come up with a intersection point.

The rest of the paper will be structured as follows. In Section 4.2, the conventions for the signals at different microphones will be laid out, as well as the way

in which the architectures predict the positions. The reference method will also be explained in more detail and then the two co-operative architectures will be shown. In Section 4.3, an evaluation of all three methods will be given. It will also be shown that the proposed methods work equally well under the condition of weakly synchronised clocks. Section 4 concludes the paper.

4.2 Models

4.2.1 Signal Model

First the conventions of this paper will be described. A microphone arrays will be used, which all consist of M microphones. The signals at microphone m of microphone array a at time sample i is the combination of J target speakers $x_{a,m,j}(i)$ and noise $v_{a,m}(i)$. $x_{a,m,j}(i)$ consists of the direct path of the speech signals as well as the reverberation of the room. In the short-time Fourier transform (STFT) domain, the microphone signals are then written as:

$$Y_{a,m}(l,k) = \sum_{j=0}^{J-1} X_{a,m,j}(l,k) \quad + V_{a,m}(l,k) \tag{4.1}$$

where k is the frequency index and l the time index of the STFT. This representation is often chosen for speaker localisation and separation due to the well-known properties of sparsity and disjointness of speech signals in the STFT representation [23].

In vector notation, the signals at microphone array a are represented by

$$\boldsymbol{Y}_a(l,k) = [Y_{a,0}(l,k), \ldots, Y_{a,M-1}(l,k)]^T \tag{4.2}$$

and the signals from all the microphone arrays are represented by

$$\boldsymbol{Y} = [\boldsymbol{Y}_0(l,k), \ldots, \boldsymbol{Y}_{A-1}(l,k)]^T. \tag{4.3}$$

4.2.2 Prediction models

The starting point of all proposed methods is the convolutional recurrent DNN proposed by Bohlender et al. [10], which extends the convolutional DNN (CNN) based approach of [9] with temporal context. Their network uses a single microphone array to estimate the direction(s) of arrival (DoA) of target speaker(s). It is assumed, here, that the target speaker is in the far field of the microphone array, meaning that the amplitude carries less information regarding the speaker location(s). That is why the input features of the neural network are the phases for all M microphones and K frequencies: $[\angle \boldsymbol{Y}_a(0,l), \ldots, \angle \boldsymbol{Y}_a(K-1,l)]$.

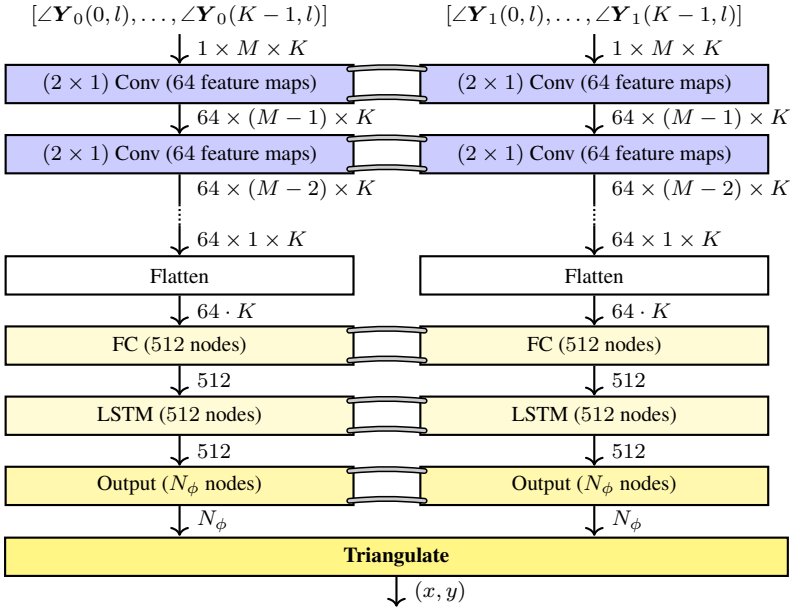


Figure 4.1: This figure shows the triangulation approach, consisting of two networks that perform DoA estimation. The grey bands represent that the weights of these layers are identical. After each convolutional layer batch normalisation is done, dropout with rate 0.5 is used before each FC layer and ReLU activation is used after each hidden layer.

To execute the 2D localisation, the newly proposed systems make use of two microphone arrays ($A = 2$), whose relative positions are known. This means that the distance between the arrays and their relative orientation are fixed. However, the position in the room and the orientation with respect to the room can be arbitrary but static. This knowledge is needed because the relation between the input features and the positions of the arrays is not linear. The localisation will be with respect to the centre of the chosen array configurations. For absolute localisation, the array positions should therefor also be known.

The phases $\angle \boldsymbol{Y}(l,k)$ are still chosen to be the input features, despite the fact that the signal amplitudes at different arrays now carry useful information. This is done in order to have a fair comparison between the triangulation and the CLAs. However, in order to still satisfy the far field assumption, combined with spatial aliasing concerns, the input features from different microphone arrays are kept separate in the first few network layers.

4.2.3 Reference method

The baseline of this paper is a triangulation approach using the DoA architecture of [10]. Two arrays are used individually to yield two independent DoA estimates.

The estimated 2D source position is the intersection point of the rays extended in the direction of the estimated DoAs from each array. The triangulation architecture is shown in Figure 4.1. This can be implemented as an almost completely distributed approach: all processing can be done on the individual microphone arrays separately, and only the two DoAs should be sent to the central processing hub which does the triangulation. This implementation has a low demand on the bandwidth to the central processor.

We present below a brief overview of the DNN used for estimating the DoA by each array. Further details may be found in the original base paper [9], and [10] for the temporal context. First, convolutional (conv) layers are used to combine information from different microphones. This is done in multiple layers, where each layer only combines information form two neighbouring microphones at a time. At this point, each frequency is processed separately. The output of the convolutional layers is then put through a fully connected (FC) layer followed by a recurrent layer: a long short-term memory (LSTM) layer. The FC layer is the first layer which has information from all frequencies. After the LSTM, the output layer classifies in which of the N_ϕ angular sectors the source lies. More output classes can allow for a finer DoA estimate. This is done for the two arrays separately. Triangulation then gives the intersection point at which the reference method estimates the location of the target speaker.

4.2.4 Proposed methods

This paper proposes to increase the localisation accuracy by mixing information from both arrays within the DNN. However, for practical implementations, we need to ensure that there are no strict synchronisation requirements between the two arrays. This is done by combining the information at a deeper layer of the DNN instead of at the input feature level. This omits the need for asynchronous training data. Two co-operative localisation architectures are discussed: narrowband mixing CLA (NM-CLA) and broadband mixing CLA (BM-CLA). The narrowband variant combines the information of the different microphone arrays right after the last convolutional layer of both arrays, doing an inter-array convolution instead of an intra-array convolution. Each frequency bin up until this point is still considered separate, which explains the naming choice.

The output is also defined as a classification problem, where each class represents a rectangular region in 2D space. The output of the DNN may be interpreted as the probability of the speaker being present within that region. The number of classes depends on how precisely we want to localise the speaker. the total number of classes is given by $N_x \cdot N_y$ where N_x represents the number of regions in the x direction and N_y that of the y direction. In Figure 4.2, the described architecture is shown.

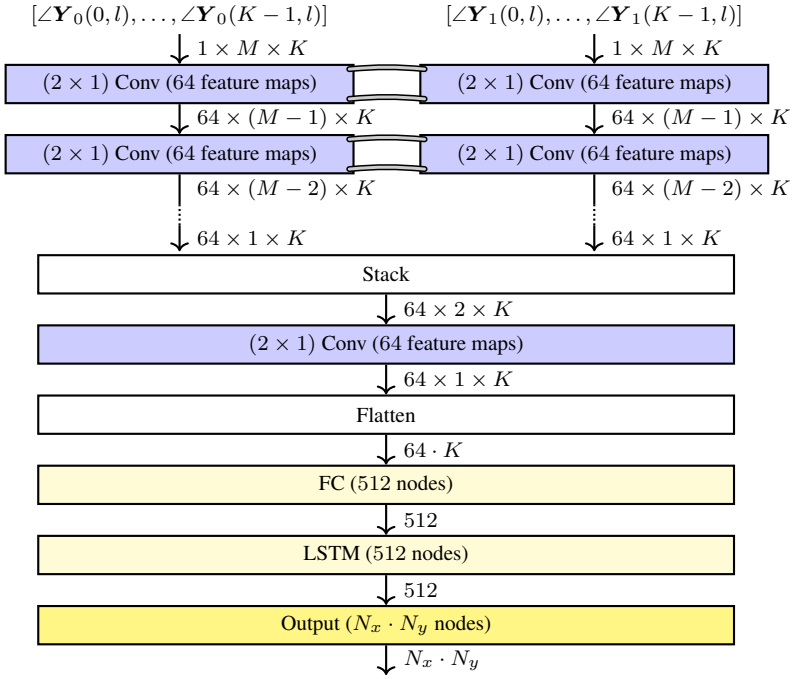


Figure 4.2: The narrowband mixing co-operative localisation architecture (NM-CLA). The architecture from Figure 4.1 is altered to mix information of both arrays right after the intra-array convolutional layers, by performing an inter-array convolution. This is narrowband in the sense that the frequency dimension is still present at that time.

This NM-CLA however does increase the minimum bandwidth requirements substantially: each node has to send $64 \cdot K$ features to the central node. In this work, the STFT has $K = 257$ frequency bins.

The second proposed variant, the BM-CLA, lowers the bandwidth requirements compared to NM-CLA. In this case, there is an intra-array convolutional layer after the FC layer. This FC layer already combines the different frequency bins, which makes the newly added convolutional layer work on broadband information. As an alternative, it was also tested to use a FC layer instead. The stack operation should then also be changed to a flatten operations, where the output dimension is then 1024. The convolutional layer was empirically found to perform slightly better. The output of the broadband architecture is again a classifier with $N_x \cdot N_y$ classes. This architecture only needs to send 512 features per microphone array to the central unit. The BM-CLA is depicted in Figure 4.3. Both CLAs do not increase the amount of trainable parameters with that much, since both of them only add one convolutional layer, which is small compared to the FC layer already present in the triangulation networks. The amount of parameters are 10.6×10^6

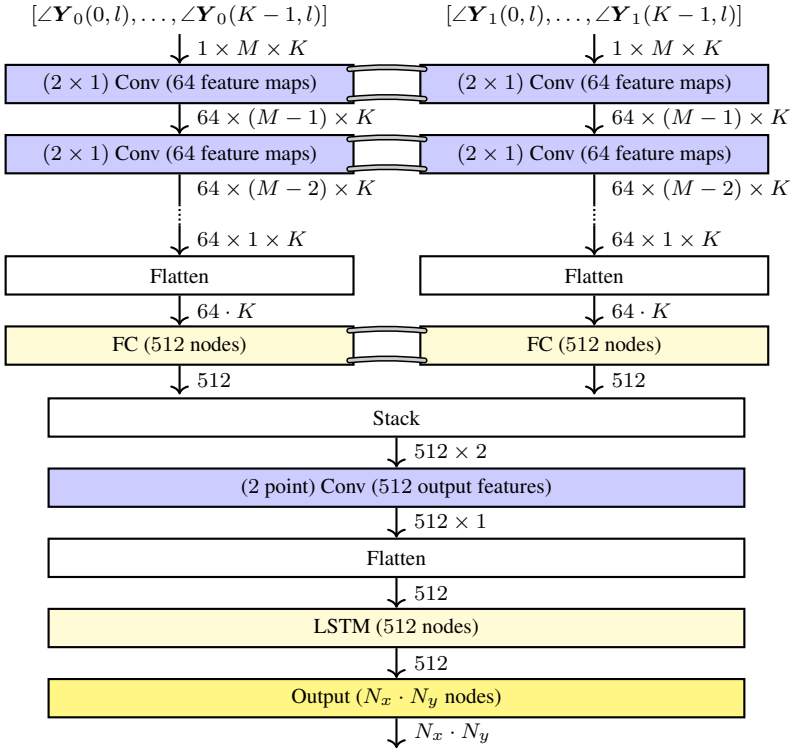


Figure 4.3: the broadband mixing co-operative localisation architecture (BM-CLA). Here the mixing of information happens after the FC layer, and is also performed by an inter-array convolution.

for triangulation, 10.7×10^6 for NM-CLA and $11,2 \times 10^6$ for BM-CLA.

In a WASN, the clocks of different nodes (here microphone arrays) cannot be assumed to be perfectly synchronised. Triangulation approaches are inherently robust against asynchronous nodes. For the proposed architectures, this needs to be verified. Section 4.3 will also present some results where asynchronicity is simulated.

4.3 Evaluation

4.3.1 Training

The training data set is generated in a similar manner as described in [10]. We want to account for time-variant source activity. This includes moving sources, or speakers becoming inactive and new speakers becoming active. A Markov model, $A_j(t) \in \{0,1\}$, is used to generate this dynamic setting. $A_j(t)$ indicates if source

number j is active ($A_j(t) = 1$) or not ($A_j(t) = 0$). The training data set contains both instances where $J \in \{0, 1, 2\}$. The average Markov transition probability between the two states is set so that there is on average one transition per 1.5s. Each time a source becomes active, it is assigned a random 2D position within the room to simulate source movement. The source signals are randomly sampled speech signals coming from the TIMIT [24] or PTDB-TUG [25] databases.

The training set consists of 10 different room dimensions paired with their own reverberation times (RT60). All the room impulse responses are generated using pyroomacoustics [26]. Spatially diffuse and temporally uncorrelated noise is added with SNRs ranging from 0 dB till 30 dB. During training, the Adam optimiser is used to minimise the binary cross entropy cost function. The batch size is chosen to be 20, where each sample consists of a sequence generated by the Markov model of length $2\,s$.

4.3.2 Evaluation

For the evaluation, two microphone arrays are used that are spaced 2 meters apart. Both arrays have four microphones, placed at each corner of a square of side 21mm. The output of the model can classify $N_x \cdot N_y$ different locations, where for this evaluation $N_x = N_y = 16$ is chosen. Each class represents a square of $0.5\,m \times 0.5\,m$, where the network output should ideally be 1 for the class where the true source position lies. For our configuration this restricts the true source position to a maximum of $4\,m$ from the centre of the two arrays in the x and y directions.

The evaluation is carried out in unseen room dimensions and random reverberation times between 0.2 s and 0.8 s. 6 SNR levels are simulated between -5 dB till 20 dB, in steps of 5 dB. For each SNR and architecture, 1000 simulations are carried out. The results are divided in two subsets: one with lower reverberation times (RT60 $< 0.5\,s$), and one with higher reverberation times (RT60$\geq 0.5\,s$). For the evaluation, we focus on the case where only one source is active.

The results can be seen in Figure 4.4. A localisation is deemed successful if the estimate is within one meter of the real speaker location. The figure clearly shows that both proposed CLAs have significantly higher localisation accuracies at all the SNR levels compared to the reference triangulation approach. This trend is even greater in the highly reverberant and high SNR case. One specific case where the CLAs outperform triangulation substantially, is where the source and the microphone arrays lie on the same line. The intersection point can then be non existing, even without error in the DoA estimates, due to their discrete nature.

Another interesting result is that BM-CLA outperforms NM-CLA in accuracy at every SNR. This in combination with the lower bandwidth requirements makes BM-CLA superior to NM-CLA. This is somewhat surprising, since we hypothe-

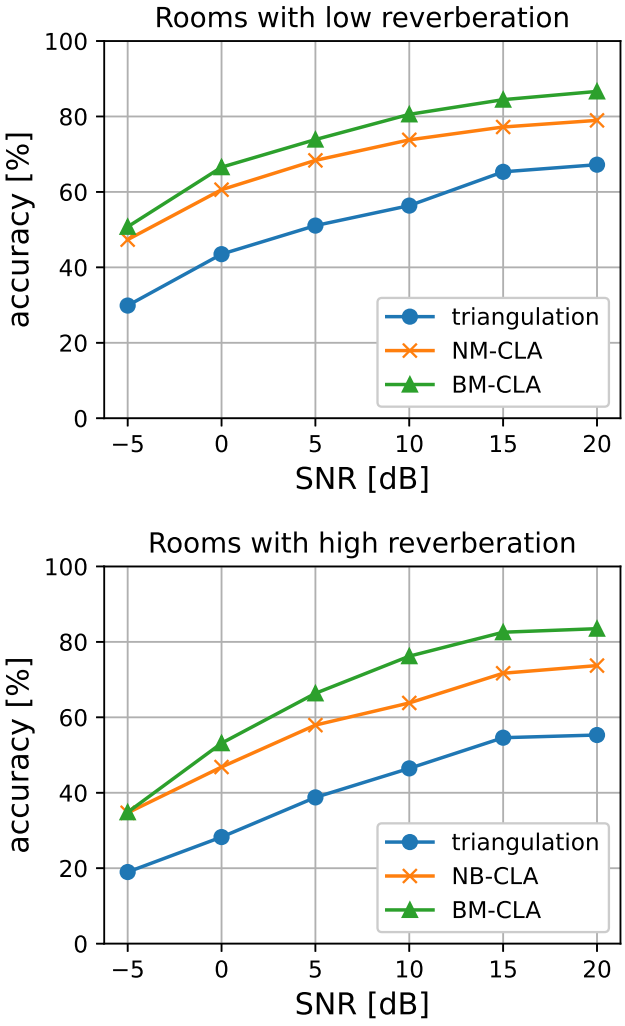
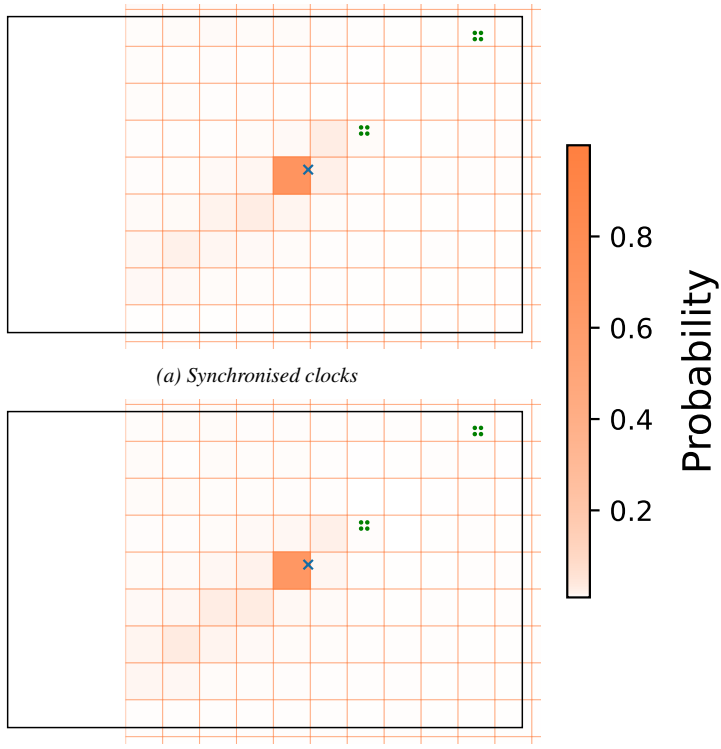


Figure 4.4: Accuracy of the localisation approaches for different SNRs. A position is considered to be correctly estimated if it is within one meter of the actual source position. The first figure shows the accuracy for low reverbrant rooms (RT60 < 0.5 s). The second figure shows the accuracy of the highly reverberant rooms.



(a) Synchronised clocks

(b) Asynchronous clocks

Figure 4.5: A depiction of the broadband mixing co-operative source estimation. The first figure is the case where the network is perfectly synchronised. In the second figure, the clock signals of both arrays differ by 10 samples. The big black box represents the room. The green dots are single microphones: two microphone arrays with 4 microphones each, are present. The blue cross is the true source position. The orange boxes are the possible areas where the co-operative DNNs can localise the source (only locations within the room are depicted). The opaqueness indicates how high the probability is that the network has given to the corresponding location.

sised that microphone arrays sharing more information would increase the accuracy. One possible explanation could be that combining the narrowband information between microphone arrays, can increase the risk of spatial aliasing since the arrays are far apart. In the broadband variant, the DNN is already forced to mix all the frequencies before the information from different arrays are combined, which reduces the spatial aliasing problem.

4.3.3 Robustness against clock asynchronicity

In order to be able to deploy the proposed systems, they should also be robust against a slight misalignment of the clock signals as it is hard to perfectly synchronise two different nodes in WASNs. The evaluation process is done by repeating the experiment from Section 4.3.2, where subsample delays are added to the RIRs of the second microphone array. The delays are uniformly sampled from 0 to 2 samples. The accuracy plots with this added asynchronicity are almost identical to those of Figure 4.4, indicating that the proposed methods are inherently robust to sampling inaccuracies between the nodes. Instead of showing this result, therefore, we consider a specific case from BM-CLA with and without perfect synchronicity which are shown in Figure 4.5. The room is 7 by 4.3 m, and 2.6 m high. The RT60 is 0.72 s and diffuse noise is added 15 dB SNR. Here the clock of the second array is actually 10 samples apart from the first array. It is clear that in both cases, the network gives the highest probability to the same (correct) position. Similar results are present for different scenarios.

4.4 Conclusion

This paper showed that the localisation accuracy in a WASN can be significantly increased by sharing information early between microphone arrays, compared to triangulation. This is done by expanding upon a convolutional recurrent DNN based approach for DoA estimation. Two different co-operative multi-array localisation methods were discussed and compared: NM-CLA and BM-CLA. NM-CLA mixes information between microphones where narrowband information is still present, while BM-CLA only does the inter-array mixing after the broadband information has already been mixed by the individual microphone arrays. BM-CLA has the best accuracy and also has a lower bandwidth requirement on the WASN. Both CLAs are robust against small deviations in clock synchronicity between different nodes. This comes inherently since the features between nodes are only mixed at deeper stages of the DNN, where they are more abstract and no longer dependent on the exact clock samples. Future work includes extending the proposed methods for use in ad-hoc WASN applications, including amplitude information for an even greater localisation accuracy and doing mask based source separation, similar to, e.g. [27].

Acknowledgment

This work is supported by the Research Foundation - Flanders (FWO) under grant numbers G081420N and 11G0721N.

References

[1] A. Bertrand. *Applications and trends in wireless acoustic sensor networks: A signal processing perspective*. In 2011 18th IEEE symposium on communications and vehicular technology in the Benelux (SCVT), pages 1–6. IEEE, 2011.

[2] A. Bertrand, J. Callebaut, and M. Moonen. *Adaptive distributed noise reduction for speech enhancement in wireless acoustic sensor networks*. In Proc. of the International Workshop on Acoustic Echo and Noise Control (IWAENC), 2010.

[3] S. Markovich-Golan, A. Bertrand, M. Moonen, and S. Gannot. *Optimal distributed minimum-variance beamforming approaches for speech enhancement in wireless acoustic sensor networks*. Signal Processing, 107:4–20, 2015.

[4] F. Alías and R. M. Alsina-Pagès. *Review of wireless acoustic sensor networks for environmental noise monitoring in smart cities*. Journal of sensors, 2019, 2019.

[5] M. Cobos, F. Antonacci, A. Alexandridis, A. Mouchtaris, and B. Lee. *A survey of sound source localization methods in wireless acoustic sensor networks*. Wireless Communications and Mobile Computing, 2017, 2017.

[6] U. and Heute and C. Antweiler. *Advances in digital speech transmission*. John Wiley & Sons, 2008.

[7] P. Pertilä and E. Cakir. *Robust direction estimation with convolutional neural networks based steered response power*. In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6125–6129. IEEE, 2017.

[8] Z.-Q. Wang, X. Zhang, and D. Wang. *Robust speaker localization guided by deep learning-based time-frequency masking*. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 27(1):178–188, 2018.

[9] S. Chakrabarty and E. A. P. Habets. *Multi-Speaker DOA Estimation Using Deep Convolutional Networks Trained With Noise Signals*. IEEE Journal of Selected Topics in Signal Processing, 13(1):8–21, 2019. doi:10.1109/JSTSP.2019.2901664.

[10] A. Bohlender, A. Spriet, W. Tirry, and N. Madhu. *Exploiting Temporal Context in CNN Based Multisource DOA Estimation*. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2021.

[11] H. Sundar, W. Wang, M. Sun, and C. Wang. *Raw waveform based end-to-end deep convolutional network for spatial localization of multiple acoustic sources*. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 4642–4646. IEEE, 2020.

[12] T. Gburrek, J. Schmalenstroeer, A. Brendel, W. Kellermann, and R. Haeb-Umbach. *Deep neural network based distance estimation for geometry calibration in acoustic sensor networks*. In 2020 28th European Signal Processing Conference (EUSIPCO), pages 196–200. IEEE, 2021.

[13] A. Brendel and W. Kellermann. *Distributed source localization in acoustic sensor networks using the coherent-to-diffuse power ratio*. IEEE Journal of Selected Topics in Signal Processing, 13(1):61–75, 2019.

[14] D. Cherkassky, S. Markovich-Golan, and S. Gannot. *Performance analysis of MVDR beamformer in WASN with sampling rate offsets and blind synchronization*. In 2015 23rd European Signal Processing Conference (EUSIPCO), pages 245–249. IEEE, 2015.

[15] W. Su and I. F. Akyildiz. *Time-diffusion synchronization protocol for wireless sensor networks*. IEEE/ACM transactions on networking, 13(2):384–397, 2005.

[16] Y.-C. Wu, Q. Chaudhari, and E. Serpedin. *Clock synchronization of wireless sensor networks*. IEEE Signal Processing Magazine, 28(1):124–138, 2010.

[17] T. Gburrek, J. Schmalenstroeer, and R. Haeb-Umbach. *Iterative Geometry Calibration from Distance Estimates for Wireless Acoustic Sensor Networks*. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 741–745. IEEE, 2021.

[18] A. Plinge, F. Jacob, R. Haeb-Umbach, and G. A. Fink. *Acoustic microphone geometry calibration: An overview and experimental evaluation of state-of-the-art algorithms*. IEEE Signal Processing Magazine, 33(4):14–29, 2016.

[19] T. Gburrek, J. Schmalenstroeer, and R. Haeb-Umbach. *Geometry calibration in wireless acoustic sensor networks utilizing DoA and distance information*. EURASIP Journal on Audio, Speech, and Music Processing, 2021(1):1–17, 2021.

[20] Á. Lédeczi, G. Kiss, B. Feher, P. Volgyesi, and G. Balogh. *Acoustic source localization fusing sparse direction of arrival estimates*. In 2006 International Workshop on Intelligent Solutions in Embedded Systems, pages 1–13. IEEE, 2006.

[21] A. Griffin, A. Alexandridis, D. Pavlidi, Y. Mastorakis, and A. Mouchtaris. *Localizing multiple audio sources in a wireless acoustic sensor network*. Signal Processing, 107:54–67, 2015.

[22] S. V. Sibanyoni, D. T. Ramotsoela, B. J. Silva, and G. P. Hancke. *A 2-D Acoustic Source Localization System for Drones in Search and Rescue Missions*. IEEE Sensors Journal, 19(1):332–341, 2019. doi:10.1109/JSEN.2018.2875864.

[23] S. Rickard and O. Yilmaz. *On the approximate W-disjoint orthogonality of speech*. In 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing, volume 1, pages I–529. IEEE, 2002.

[24] J. S. Garofolo. *Timit acoustic phonetic continuous speech corpus*. Linguistic Data Consortium, 1993, 1993.

[25] G. Pirker, M. Wohlmayr, S. Petrik, and F. Pernkopf. *A pitch tracking corpus with evaluation on multipitch tracking scenario*. In Twelfth Annual Conference of the International Speech Communication Association, 2011.

[26] R. Scheibler, E. Bezzam, and I. Dokmanić. *Pyroomacoustics: A python package for audio room simulation and array processing algorithms*. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 351–355. IEEE, 2018.

[27] A. Bohlender, A. Spriet, W. Tirry, and N. Madhu. *Neural Networks Using Full-Band and Subband Spatial Features for Mask Based Source Separation*. In 2021 29th European Signal Processing Conference (EUSIPCO). IEEE, accepted for publication in 2021.

5

Robustness of ad hoc microphone clustering using speaker embeddings: evaluation under realistic and challenging scenarios

This chapter moves from distributed microphone arrays towards ad-hoc distributed individual microphones. Here, the number and position of the available microphones are not known a priori. An indication of which microphones pick up similar content can be made by clustering the microphones with a common dominant sound source. This clustering can be seen as a relative localisation method, where at the output, the exact location is still unknown, but at least some relative location information is available that indicates which microphones are close to each other. This information could in turn be exploited by separation DNNs, which will happen in Chapters 7 and 8. This chapter introduces speaker embeddings from pre trained speaker verification networks as a feature for microphone clustering. These embeddings should be particularly descriptive since goal of the speaker verification task is to produce similar embeddings for signals dominated by the same speaker, while generating very different embeddings for signals dominated by different speakers. This is evaluated with a realistically simulated dataset, and under a variety of different scenarios (closely spaced sources, quick adaptation times, ...).

⋆ ⋆ ⋆

Stijn Kindt, Jenthe Thienpondt, Luca Becker and Nilesh Madhu

Published in the EURASIP Journal on Audio, Speech, and Music Processing.

Abstract Speaker embeddings, from the ECAPA-TDNN speaker verification network, were recently introduced as features for the task of clustering microphones in ad hoc arrays. Our previous work demonstrated that, in comparison to signal-based Mod-MFCC features, using speaker embeddings yielded a more robust and logical clustering of the microphones around the sources of interest. This work aims to further establish speaker embeddings as a robust feature for *ad hoc* microphone clustering by addressing open and additional questions of practical interest, arising from our prior work. Specifically, whereas our initial work made use of simulated data based on shoe-box acoustics models, we now present a more thorough analysis in more realistic settings. Further, we investigate additional important considerations such as the choice of the distance metric used in the fuzzy C-means clustering; the minimal time range across which data need to be aggregated to obtain robust clusters; and the performance of the features in increasingly more challenging situations, and with multiple speakers. We also contrast the results on the basis of several metrics for quantifying the quality of such ad hoc clusters. Results indicate that the speaker embeddings are robust to short inference times, and deliver logical and useful clusters, even when the sources are very close to each other.

5.1 Introduction

Many 'smart' devices carry at least one microphone. Typical examples are phones, smart watches and laptops. There is also a trend towards the internet of things (IoT) and smart homes, increasing the number of microphone-carrying devices scattered around a room. Sharing information from all these microphones, by forming an acoustic sensor network (ASN), can give a good acoustic coverage of a room/living environment. This can be exploited for tasks like acoustic event detection, classification, and separation, in scenarios such as assisted living and healthcare, hearing aids, communications, etc. (see e.g., [1]).

Since the microphones can be distributed all over the room, the spatial diversity is greater than that of a compact microphone array (microphones in close proximity). However, combining the signals of such distributed microphones is not straightforward. Firstly, the microphones may not be driven by the same clock, so sample rate offsets (SROs) and sample time offsets (STOs) may be present. The

relative time delay between signals at different microphones is therefore no longer only an effect of the propagation delays. Additionally, if the ASN is connected via wireless links (WASNs), bandwidth and processing power limitations are introduced. Furthermore, for portable microphone-carrying devices, the position of the microphones is not known *a priori*, and forming an ASN from such *ad hoc* distributed microphones makes it even harder to perform localisation or separation.

In order to cope with the unknown microphone positions, it is often helpful to cluster microphones based on the similarity of the signals they capture. Thereby all microphones dominated by the same source may be expected to be grouped in the same cluster. Similar clustering of microphones which primarily pick up the ambient signal or noise can be performed. Such clustering has already been proven valuable for subsequent steps like source classification (e.g. [2, 3]) and separation (e.g. [4, 5]).

The clustering procedure consists of two main stages: (i) proper selection of acoustic features, upon which clustering is carried out, and (ii) choosing an appropriate clustering algorithm. Below we first discuss prior work in this regard, before outlining the main contributions of our work.

5.1.1 Prior work

A variety of clustering features have been proposed in the literature. For example, the magnitude squared coherence (MSC) between microphones on the noise-only part of the signal is used in [4]. Assuming the noise field to be diffuse gives a direct relation between the noise-MSC and the inter-microphone distance. In a similar vein, the room impulse responses (RIRs) are first estimated for each microphone in [6], and are subsequently used to cluster microphones. Such classes of techniques depend solely on the room properties to perform clustering.

In contrast, the MSC on the *speech-active* parts of the signal is utilised as cluster features in [3]. This contains information about the RIRs and the *content* of the signal, thus both the room characteristics and signal correlation is exploited. Similarly, in [7], the individual microphone auto-correlation of the source signal and the auto-correlation of the noise signal are computed, where identification of the noise and source regions is done with the help of voice activity detection (VAD). These yield source- and location-specific features, which are used for the clustering.

All the above-mentioned techniques are influenced by the room characteristics. These characteristics could be useful if geometry-related information is required, *e.g.*to estimate the position of the microphones in the room. This would however also require a prior calibration stage for different positions in the room, as done in [8]. In contrast, features that are *speech*- or *content*- specific are useful to be able to focus on pre-determined targets (*e.g.*in care homes, where monitoring of

particular patients may be desired). Additionally, *speaker-specific* features can lead to a more targeted clustering, and without the need to first estimate the room-acoustics.

Clustering based on purely signal-dependent features has, therefore, also been investigated. The work in [2, 5, 9, 10] proposed hand-crafted features, based on the modulation-domain Mel frequency cepstral coefficients (Mod-MFCCs), where mean subtraction reduces the effect of the room characteristics under the assumption that the source and microphone stay sufficiently static. In contrast, the work in [11] depends on data-driven feature extraction, where a variational auto-encoder (VAE) trained on all types of speech and music data is used within a federated learning framework. After training, the parameters of the bottleneck layer are randomised and the model is distributed to all the microphone nodes. During runtime, each node updates the bottleneck weights based on the captured signal, essentially overfitting on that signal. The accumulated gradients from multiple rounds of back-propagation are sent back to the central node and are used as cluster features. The advantage of this approach is the privacy preservation of the speaker. However, the privacy constraint inevitably precludes the use case where speaker-specific processing is desired. Also, retraining the network at each node comes at a relatively high computational cost, which has been discussed and improved in [12].

Since the primary goal is to detect and cluster microphones around speech sources, we introduced speaker embeddings – representation of a talker in a high-dimensional *latent space* – as features in [13]. The embeddings are generated by a pre-trained speaker verification network: the Enhanced Propagation and Aggregation Time Delay Neural Network (ECAPA-TDNN) [14]. Since speaker verification should be robust to different room characteristics and perturbations, the embedding network is trained with appropriately augmented data, yielding room-independent and yet source-specific embeddings, which serve well as clustering features.

For the clustering algorithm itself, we note that there are many approaches in the literature, e.g., K-means is used in [6], non-negative matrix factorisation is utilised in [3], while matrix bi-partitioning is deployed in [11]. In contrast, fuzzy C-means (FCM) is incorporated in approaches based on the mod-MFCC features [2, 5, 10]. The fuzzy weights indicate the *degree* to which a microphone belongs to a cluster - which is indirectly an indication of the strength of the target source at that microphone. Therefore, we also adopted FCM in our approach, as the fuzzy weights can be informative for later stages, like enhancing the source.

This work builds on the initial results of [13]. The goal is to obtain a holistic overview of the opportunities and limitations of using speaker embeddings as clustering features for *ad hoc* distributed microphones. The main contributions of this work are outlined below.

5.1.2 Contributions

For the FCM clustering, the standard Euclidean distance was used in previous work, whereas speaker verification implementations typically use the cosine similarity, as the direction and orientation of the embeddings yield more discrimination. Therefore, as part of this work, we investigate the benefit of using the cosine distance in FCM-based clustering.

Also, our initial comparison [13] of the speaker embeddings with the Mod-MFCC based features was in simulated shoe-box rooms. There it was shown that the embeddings generate more robust and visually logical clusters. It was also assumed that the sources to be separated were sufficiently far apart and the feature extraction was on data aggregated across a relatively long time-span of 4s. This initial study raised several interesting questions, which are handled in the current contribution, namely: (i) what is the effect of realistic room environments on the features? (ii) As mentioned above, what distance metric is best suited for the clustering? (iii) What happens if the sources were placed in close proximity? (iv) Does the time-scale of data aggregation affect the performance? And, last but not least, (v) can speaker-embedding-based features be used to detect the presence of known talkers and only extract them in realistic, *dialogue-like* situations? We believe that answering these questions is important to obtain a full picture for practical implementations.

For realistic room environments, we employ the SINS database [15]. We systematically evaluate the performance on distant- and closely-spaced sources. Next, we vary the duration of the segment on which the clustering features are generated. The former will generate insights into the robustness of the features under increased difficulty, while the latter indicates the feasibility of adapting to quickly changing environments (more frequent updates on shorter segments) or of scaling the complexity (e.g., for bandwidth and power constraints) by updating less frequently and on shorter segments.

For quantitative appreciation of the results, a concept of cluster quality needs to be defined. However, this is not a trivial task, as generating the ground truth is not straightforward. Thus, we proposed three intuitive metrics in [13]: (I) the histograms of the direct-to-reverberant and (II) direct-to-reverberant-interference-and-noise ratios (DRR and DRINR) of microphones attributed to a speech-source cluster (indicating the quality of the microphones allocated to a cluster), and (III) the average number of microphones in a speech-source cluster (indicating spatial diversity available at a cluster). Additionally, we also benchmark on cluster-based speaker separation from [5]. With these metrics taken together, we obtain a more holistic performance overview.

The rest of the paper is structured as follows: in Section 5.2 we will write out the signal model followed by a succinct explanation of the Mod-MFCC and speaker embedding based features in Section 5.3. The FCM algorithm is discussed

in Section 5.4, followed by the speaker separation scheme for evaluation in Section 5.5. Section 5.6 explains the different situations we evaluate, as well as the metrics we use to benchmark the clustering. The discussion of the results is done in Section 5.7, and Section 5.8 concludes the paper.

5.2 Signal model

For our setup, we consider J concurrently active sources and M microphones distributed in the room. The m-th microphone signal, y_m, is given as:

$$y_m(n) = \sum_{j=1}^{J} x_{j,m}(n) + v_m(n), \tag{5.1}$$

where n is the discrete time index, $x_{j,m}$ is the source signal captured by the m-th microphone and generated by the j-th source, and v_m symbolises the additive noise at the m-th microphone.

In the following, we shall use the short-time Fourier transform (STFT) representation of the signal for processing. The signal in this domain is denoted as:

$$Y_m(l,k) = \text{STFT}\big[y_m(n)\big], \tag{5.2}$$

where k is the STFT frame index and l is the index of the discrete frequency bin.

5.3 Clustering features

As previously mentioned, there are three major categories of feature types on which clustering has been performed. The first set, based on estimating the relative locations of the microphones with respect to each other, is termed geometry-based features (GBFs). The second class of features exploits geometry and signal information and is termed as signal-based features (SBFs). The last set generates features that are source-specific and we term these source-dependent latent features (SDLFs).

GBFs extract information relating to the relative spatial distances between microphones. This can be obtained explicitly by estimating the RIRs ([6, 8]) or implicitly, using the coherence in the noise-only periods as in [4].

SBFs are computed by comparing signals across different microphones and typically contain information on the acoustic environment and the source signals. The use of the MSC, as in [3, 7] are examples of such feature usage.

The use of SDLFs is based on the fundamental assumption that signals from microphones close to the same source will generate similar *latent* features. Additionally, if the latent features are designed to be source discriminating, features

characterising one source should be very different from those for other sources and the ambient noise. A seminal example here is the set of hand-crafted Mod-MFCC features proposed in [9]. A data-driven approach to get SDLFs is proposed in [11], which is based on the use of auto-encoders and federated learning principles.

Although the latter two methods try to focus on the source-specific characteristics, there will always be some influence of the room characteristics – which reduces the discriminative capacity of these features. Therefore, we propose to use speaker verification networks to generate source-specific features, as these networks are trained to generate the same embedding for a speaker with relative robustness to the environmental conditions. Additionally, as the embeddings should be sufficiently unique in order to discriminate between *different* speakers, they can yield a robust indication of source dominance at a microphone – making them ideal for the application to *ad hoc* arrays.

Given our focus on demonstrating the benefits of source-specific features in ASNs, we limit ourselves to SDLFs in this study. Specifically, we use the Mod-MFCC features as a baseline for benchmarking speaker embedding features. The federated learning framework is not considered due to its large computational cost and complexity (multiple rounds of backpropagation are needed). Furthermore, in contrast to speaker embeddings, information about specific talkers cannot be exploited within this framework - making it less versatile.

5.3.1 MFCC-based features

The modulated Mel-frequency cepstral coefficients (Mod-MFCC) based features were first utilised in [2, 9]. These hand-engineered features consist of two $\mathcal{N}$-dimensional cepstral modulation ratios (CMR) and one $\mathcal{N}$-dimensional averaged modulation amplitude (AMA), where $\mathcal{N}$ is the number of considered cepstrum bins.

We briefly summarise the computation of these features as proposed in [2], and subsequently denoted as $\boldsymbol{\mathcal{F}}^{\text{MFCC}}$. First, the MFCC, $Y_{\text{MFCC}}(\eta, k)$ are computed from the STFTs in (5.2). Here η is the cepstral index. Cepstral mean subtraction (CMS) is applied to reduce the effect of reverberation, resulting in features that better capture the speech structure [16, 17].

$$\tilde{Y}_{\text{MFCC}}(\eta, k) = Y_{\text{MFCC}}(\eta, k) - \frac{1}{K} \sum_{k=0}^{K-1} Y_{\text{MFCC}}(\eta, k)\,. \tag{5.3}$$

The Mod-MFCC is then calculated as the DFT of the MFCC features with a rectangular window of length L:

$$Y_{\text{Mod-MFCC}}(\kappa, \eta, \lambda) = \sum_{l=0}^{L-1} \tilde{Y}_{\text{MFCC}}(\eta, \lambda Q + l) e^{-j2\pi l\kappa/L}\,, \tag{5.4}$$

where $\lambda \in \{0, \cdots, \Lambda - 1\}$ is the modulation index, Q the modulation shift and $\kappa \in \{0, \cdots, L/2\}$ is the modulation frequency bin. Averaging the modulation amplitude spectra, $|Y_{\text{Mod-MFCC}}(\kappa, \eta, \lambda)|$, over time is done in order to be robust against time shifts that are expected in ASNs:

$$\hat{Y}_{\text{Mod-MFCC}}(\kappa, \eta) = \sum_{\lambda=0}^{\Lambda-1} |Y_{\text{Mod-MFCC}}(\kappa, \eta, \lambda)| \,. \tag{5.5}$$

Then the Cepstral Modulation Ratios (CMR) features and averaged modulation amplitude (AMA) features are defined as:

$$\text{CRM}_{\kappa_1|\kappa_2}(\eta) = \frac{\sum_{\kappa=\kappa_1}^{\kappa_2} \hat{Y}_{\text{Mod-MFCC}}(\kappa, \eta)}{(\kappa_2 - \kappa_1 + 1)\hat{Y}_{\text{Mod-MFCC}}(0, \eta)} \,, \tag{5.6}$$

$$\text{AMA}(\eta) = \frac{1}{L/2 + 1} \sum_{\kappa=0}^{L/2} \hat{Y}_{\text{Mod-MFCC}}(\kappa, \eta) \,. \tag{5.7}$$

The final MFCC-based feature vector is then: $\boldsymbol{\mathcal{F}}^{\text{MFCC}} = [\mathbf{AMA}^T, \mathbf{CRM}_{1|1}^T, \mathbf{CRM}_{2|8}^T]^T$, where $\mathbf{AMA}$, $\mathbf{CRM}_{1|1}$ and $\mathbf{CRM}_{2|8}$ are $\mathcal{N}$-dimensional column vectors. The first cepstral bin is omitted ($\eta \in \{1, \cdots, \mathcal{N}\}$) to reduce sensitivity to the amplitude of the signals.

5.3.2 Speaker verification-based features

Speaker embeddings refer to the representation of a talker in a high-dimensional latent space. In speaker verification tasks, such embeddings are used to test if two audio utterances are spoken by the same person. For this, embeddings extracted from the utterances are compared using a similarity metric that is appropriate to the embedding extractor architecture. The utterances are accepted as coming from the same speaker if the similarity exceeds a predetermined threshold. Applied to our case, such embeddings, extracted from the individual microphone signals, can similarly be compared – whereby microphones dominated by the same speaker would yield embeddings that are near identical.

The embedding features are generated by the recent Emphasized Channel Attention, Propagation and Aggregation Time Delay Neural Network (ECAPA-TDNN) [14]. ECAPA-TDNN improves upon the popular x-vector architecture [18] by introducing several enhancements. First, an attentive statistics pooling layer is incorporated into the network which emphasises important frame- and channel-level features during the statistics pooling operation. Additionally, a speech-adapted version of Squeeze-Excitation (SE) [19] is introduced to inject global context in the intermediate frame-level features of the model. Finally, multi-layer feature aggregation before the pooling layer gives the model the opportunity to incorporate information learned from multiple levels in the network. The ECAPA-TDNN

model is optimised using the Additive Angular Margin (AAM) [20] softmax loss function. This enables us to also consider the cosine similarity as the similarity metric for comparing two embedding vectors. We use the same training procedure as described in [14].

The embedding features, $\boldsymbol{\mathcal{F}}^{\text{SpVer}}$, extracted for each microphone, are directly input to the clustering algorithm.

5.4 Fuzzy C-means clustering

We use, similar to Gergen *et al.*[2], the fuzzy C-means (FCM) algorithm to cluster the microphone features. FCM is closely related to the K-means algorithm, with the main difference being the fuzzy membership values (FMV) included in FCM. K-means generates hard clusters where a microphone is either part of the cluster or not. However, the FMVs, which reflect how much a microphone belongs to each cluster, are useful for subsequent processing. It can for instance be used to determine the *reference* microphone, or indicate that certain sources, although part of one cluster, also contains information about another cluster. The first is useful in estimating initial speaker separation masks [10], while the latter can reasonably increase the number of microphones to be included in beamforming efforts [5]. Additionally, the FMV can be used to inform a weighted delay-and-sum beamformer (DSB) [5]. These separation methods will be discussed in more detail in Section 5.5.

In general, we will generate $C = J + 1$ fuzzy clusters. That is, one cluster for each source and one background (noise) cluster. The background cluster ideally collects all the microphones dominated by noise or reverberations, thus assuring that each microphone from a source cluster is dominated by that source.

5.4.1 FCM algorithm

The FCM algorithm minimises the following weighed error function [21]:

$$\mathcal{L} = \sum_{c=0}^{C-1} \sum_{m=0}^{M-1} \mu_{m,c}^{\alpha} \delta(\boldsymbol{\mathcal{F}}_m, \boldsymbol{\mathcal{C}}_c) \tag{5.8}$$

where $\mu_{m,c}$ are the FMVs, $\delta(\boldsymbol{\mathcal{F}}_m, \boldsymbol{\mathcal{C}}_c)$ is the distance metric between the features of microphone m and the c-th cluster centre $\boldsymbol{\mathcal{C}}_c$, and α is the fuzzy weighting exponent. Putting α to 1 will result in hard clusters, while setting $\alpha \to \infty$ will results in $\mu_{m,c} \to 1/C$; thus a bigger α will result in fuzzier clusters. Typically $1 \leq \alpha \leq 2$.

The minimisation of (5.8) is accomplished by iteratively updating the cluster

centres and FMVs with the following functions:

$$\boldsymbol{\mathcal{C}}_c = \frac{\sum_{m=0}^{M-1} \mu_{m,c}^{\alpha} \boldsymbol{\mathcal{F}}_m}{\sum_{m=0}^{M-1} \mu_{m,c}^{\alpha}} \tag{5.9}$$

$$\mu_{m,c} = \left(\sum_{\tilde{c}=0}^{C-1} \left(\frac{\delta(\boldsymbol{\mathcal{F}}_m, \boldsymbol{\mathcal{C}}_c)}{\delta(\boldsymbol{\mathcal{F}}_m, \boldsymbol{\mathcal{C}}_{\tilde{c}})} \right)^{2/(\alpha-1)} \right)^{-1} \tag{5.10}$$

5.4.2 Distance metrics

Whereas previous works primarily used the standard Euclidean distance metric:

$$\delta_{\text{Euclid}}(\boldsymbol{\mathcal{F}}_m, \boldsymbol{\mathcal{C}}_c) = ||\boldsymbol{\mathcal{F}}_m - \boldsymbol{\mathcal{C}}_c||_2^2 \tag{5.11}$$

where $\|.\|_2$ is the ℓ_2 norm of a vector, we investigate, here, the cosine distance as well:

$$\delta_{\text{Cos}}(\boldsymbol{\mathcal{F}}_m, \boldsymbol{\mathcal{C}}_c) = 1 - \frac{\boldsymbol{\mathcal{F}}_m^T \boldsymbol{\mathcal{C}}_c}{||\boldsymbol{\mathcal{F}}_m||_2 ||\boldsymbol{\mathcal{C}}_c||_2} \,. \tag{5.12}$$

This choice of similarity metric also derives from work on speaker verification. For speaker embeddings extracted by the ECAPA-TDNN, the closeness of two embedding vectors is related chiefly to their *direction and orientation* because of the AAM loss function used. In a similar manner, since the mod-MFCC features should ideally be scale-invariant, the cosine distance is applicable here as well and, as we demonstrate, turns out to be more discriminative.

5.5 Cluster-based source separation

The separation framework used here is *identical* to that described in [5, 10]. The main steps are as follows: first, we obtain an initial estimate of the target source in each cluster by means of time-frequency masking (Section 5.5.1). These initial estimates are then used to time-align the microphone signals in the respective clusters. Following, a simple delay-and-sum beamforming (DSB) is applied to compute the enhanced target signal for the cluster (Section 5.5.2). Additionally, the fuzzy membership values will be exploited to perform a weighted delay-and-sum beamformer, termed fuzzy membership value aware DSB (FMVA-DSB) (Section 5.5.3). As the last step, the improved source estimates are used to compute a postfilter (Section 5.5.4), which is applied to the beamformed signals for additional noise and interference suppression. These steps are schematically depicted in Figure 5.1

While this is a relatively simple framework, it is still insightful because the quality of the speaker separation is directly correlated with the cluster quality. Additionally, it allows a straightforward possibility to include the fuzzy membership values within the framework – which gives more insight into the clustering. This

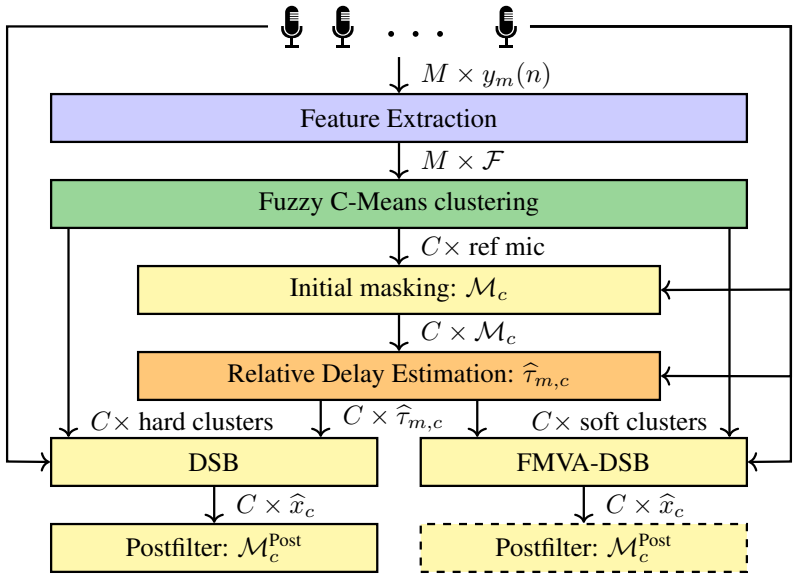


Figure 5.1: Scheme for clustering and cluster based source separation. Features (either Mod-MFCCs or speaker embeddings) extracted from the microphone signals are used to cluster the microphones. Inter- and intra-cluster information is then exploited to extract the sources dominant in each speech cluster. Yellow blocks indicate stages at which speaker separation can be performed – and which we use for evaluation. These consist of initial masking, delay and sum beamforming (DSB), fuzzy membership value aware DSB (FMVA-DSB) and postfiltering one of the DSB outputs. The dotted box is a condition that is not included in the tabulated results.

makes the framework a good tool for evaluating the clustering. Note that this does not gainsay the importance of more sophisticated methods, e.g., using cross channel correlations [22, 23] to statistically optimise the separation. Only, this is not fully relevant to the scope of the current study (improving the clustering), and can be tackled in future work. We can reasonably expect a good clustering to improve the performance of the more sophisticated methods as well.

5.5.1 Initial source estimation

The time-frequency (T-F) masks – $\mathcal{M}(l,k)$ – used for the initial estimate are obtained based on the empirically validated assumption that localised speech sources are approximately W-disjoint in their STFT representation [24]. In order to compute this mask, we assume the amplitude at T-F bins from microphones close to the target sources is greater than the amplitude of the microphones close to other sources or the background microphones. Thus, if we choose a reference for each source, we can compare their amplitudes to obtain a *rough* indication of which T-F bins are dominated by which source. By including information from the reference microphone allocated to the background cluster additionally helps to suppress reverberation and noise in the initial estimate.

We can directly use the FMVs to select the reference microphone $Y_c^{\text{ref}}(l,k)$ of each cluster c. This is simply done by selecting the microphone with the highest fuzzy value for that cluster:

$$Y_c^{\text{ref}}(l,k) = Y_m(l,k) \text{ if } \mu_{m,c} > \mu_{\bar{m},c}\,, \quad \forall \bar{m} \in \{0,\cdots,M-1\}, \bar{m} \neq m \tag{5.13}$$

Now that we have chosen a reference signal for each cluster, the respective binary mask, $\mathcal{M}_c(l,k)$, is obtained by comparing the amplitude of each T-F bin of the reference signals:

$$\mathcal{M}_c(l,k) = \begin{cases} 1 & |Y_c^{\text{ref}}(l,k)| > \dfrac{1}{B}\sum_{b=l-B+1}^{l} |Y_{\bar{c}}^{\text{ref}}(b,k)|, \\ & \forall \bar{c} \in \{0,\cdots,C-1\}, \bar{c} \neq c \\ 0 & \text{else}\,. \end{cases} \tag{5.14}$$

Here, we have introduced the averaging parameter B which, while not required for conventional binary masking, is necessary for the ASN setting. This is because the inter-microphone delay for a source is non-negligible compared to the STFT length and frameshift due to the much larger microphone spacings. These delays induce jitter in the STFT amplitudes, and consequently would do the same to the masks without averaging.

The obtained masks of a cluster c can then be applied to the microphone signals to get the source estimate $\widehat{X}_{m,c}^{\text{Mask}}(l,k)$ of that cluster:

$$\widehat{X}_{m,c}^{\text{Mask}}(l,k) = \mathcal{M}_c(l,k)\, Y_m(l,k) \tag{5.15}$$

5.5.2 Mask-based delay-and-sum beamforming

The mask can already extract the corresponding source from the mixture at each microphone. However, masks are inherently non-linear operations and combined with the crude definition of the initial mask results in sub-par quality and intelligibility of the masked signals. A better signal estimate can be obtained by a simple delay and sum beamformer. In contrast to compact microphone arrays, the inclusion of more microphones does not necessarily improve the separation capability of the beamformer [4]. Therefore, only microphones with sufficient target dominance should be considered – and this information is reflected in the FMV.

Thus, to attribute microphones to a cluster, we transform the fuzzy clusters into hard partitionings based on the FMV. A microphone m is allocated to cluster c if:

$$\mu_{m,c} > \mu_{m,\bar{c}}, \quad \forall \bar{c} \in \{0, \cdots, C-1\}, \quad \bar{c} \neq c\,. \tag{5.16}$$

We will denote the corresponding signal as $y_{m,c}$, and M_c the number of microphones in cluster c.

To compensate for the inter-microphone delays, we first have to estimate these. For this, the masks, $\mathcal{M}_c(l,k)$, are applied to all the microphone signals of the respective cluster – yielding an initial estimate of the underlying source signal of *that* cluster. The delay $\widehat{\tau}_{m,c}$ with respect to the reference microphone of cluster c is then computed from these estimates by simple correlation analysis. Time-alignment is then performed on the *unprocessed* microphone signals $y_{m,c}$, following which the DSB is computed for cluster c:

$$\widehat{x}_c^{\text{DSB}}(n) = \frac{1}{M_c} \sum_m y_{m,c}(n - \widehat{\tau}_{m,c})\,. \tag{5.17}$$

Note that the original microphone signals, and not the masked signals, are used in (5.17) since we do not want the distortions caused by the masks in the beamformer output.

5.5.3 FMV-aware delay-and-sum beamforming

As an extension to the DSB, [5] proposed a fuzzy membership value aware DSB (FMVA-DSB). This better exploits the information given by the FCM where, ideally, the microphones best capturing a source will have high FMV for that source cluster. Thus, the FMVA-DSB output is obtained by a straightforward modification of (5.17) to yield the weighted sum:

$$\widehat{x}_c^{\text{FMVA-DSB}}(n) = \frac{1}{\sum_m \mu_{m,c}} \sum_m \mu_{m,c}\; y_{m,c}(n - \widehat{\tau}_{m,c})\,. \tag{5.18}$$

Note that despite the soft weighting applied in (5.18), the $y_{m,c}$ are still only the signals of microphones that are 'hard-clustered' to cluster c.

5.5.4 Postfiltering

Similar to the initial mask, a binary mask can be computed to remove leftover interference and noise. This is particularly useful for the lower frequencies since those are hard to improve with simple beamforming. The postfilter is computed on the output of the DSB (or FMVA-DSB) as follows:

$$\mathcal{M}_c^{\text{Post}}(l,k) = \begin{cases} 1 & |\widehat{X}_c^{\text{B}}(l,k)| > \dfrac{1}{B}\sum_{b=l-B+1}^{l} |\widehat{X}_{\bar{c}}^{\text{B}}(b,k)|, \\ & \forall \bar{c} \in \{0,\cdots,C-1\}, \bar{c} \neq c \\ 0 & \text{else}\,, \end{cases} \tag{5.19}$$

where $\widehat{X}_c^{\text{B}}(l,k)$ is the STFT representation of the beamformed signal at source cluster c. This postfilter is subsequently applied to the *beamformed* signal in a similar manner to (5.15).

5.6 Experimental study

5.6.1 Focus of the study

Our prior work demonstrated the benefit of speaker embeddings for microphone clustering using simulated scenarios based on shoe-box acoustic models. This served as a proof-of-concept study, and raised the following interesting questions:

Q1. What is the clustering performance in realistic room environments (speakers of varying loudness, real room responses,...)?

Q2. What is the effect on the choice of the distance metric used in the clustering?

Q3. How does the clustering performance degrade as the sources are in closer proximity?

Q4. How does the time-scale of data aggregation affect the performance?

Q5. Given that speaker-embeddings are talker-specific, can this be exploited to detect known talkers and only extract them in realistic, *dialogue-like* situations?

These are addressed through the experimental evaluation.

5.6.2 Realistic setup - SINS database

For the evaluations, we make use of the realistic room impulse responses (RIRs) available in the SINS database [15]. This database is based upon the apartment layout and properties of the apartment used in [25], which is depicted in Figure 5.2.

As can be seen, there is a big living area (with an open kitchen), a bedroom, a bathroom, a toilet and a hall. The total floor area is $50m^2$. CATT-Acoustic with cone-tracing [26] is used to compute the RIRs for different combinations of source and microphone positions. This is an important step towards validation of the system in real world settings, and a step up from shoe-box acoustics used in our previous work. In turn, this might validate the usefulness of shoe-box simulation as an evaluation setup if the results stay consistent!

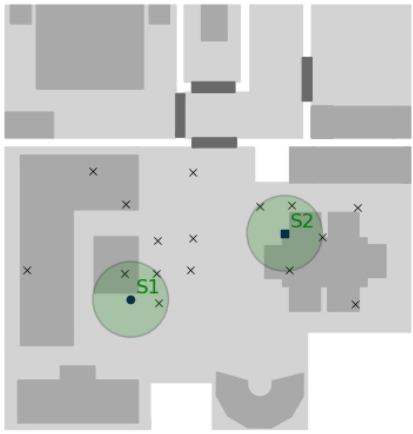


Figure 5.2: SINS room for a specific scenario. The solid dots indicate the location of the two sources, while the crosses are the microphone positions. The green circles indicate the critical distance region for each source ($d_{crit} = 0.68\,m$ for the room).

To better interpret the performance of the system, we split the scenarios into two sets based on the inter-source distance. The first set of scenarios – designed to answer Q1 – is a direct parallel to what we previously did using shoe box acoustics. The scenario only selects sources and microphones from within the living (and kitchen) area, where one source is in the left half of the room, and another one in the right half. For maximum interpretability, we avoid cases where the critical distance regions of the sources can overlap.

The second set of scenarios increases the difficulty of microphone-cluster assignment by bringing the sources closer to each other. In this setting, sources are separated by at *most* three times the critical distance of the room, while the minimum distance is limited by the dataset to $0.4\,m$. Since the critical distance of the room is $d_{\text{crit}} = 0.68\,m$, the critical distance regions of the sources will overlap. The performance in such situations will provide us with an answer to Q3.

In both scenarios, we shall test the Euclidean metric as well as the cosine distance metric – the point of Q2.

To answer Q4 – how the segment length for feature extraction influences the resulting clusters – we will revert to the first set of scenarios, to reduce the influence

of other factors. We will take 4sec as our baseline, consistent with prior work in the literature, and benchmark the performance here against segment lengths of $2; 1$ and 0.5 seconds.

Lastly, for Q5, we incorporate a known speaker embedding into the clustering algorithm. For this, we generate a scenario where the interfering speaker is constantly active, whereas the known speaker is active only for a short time in the middle of the scenario. Since the speaker is *known*, we initialise one cluster centre using the pre-computed speaker embedding of the known speaker. While the target source is inactive, we should ideally have an empty cluster for this source, while the cluster should be populated by microphones during the period of source activity.

For each scenario, there are 200 different settings with $M = 16$ microphones distributed across the room, and the presence of $J = 2$ sources for each setting. Further, we ensure that at least 3 microphones are picked from within the critical distance of each source, while the locations of the other $16 - 3J$ microphones are chosen at random.

The database consists of four-element microphone array nodes. Since we consider individually distributed microphones, we only pick one microphone from each node. We do, however, select a random microphone in order to increase the diversity in the scenarios.

5.6.3 Audio data

The LibriSpeech corpus [27] is chosen for the dry speech sources in the experiments. In line with previous work ([28]): signals of 10s are selected from the train-clean-100 LibriSpeech subset, where a voice activity detector is used to verify the presence of speech in the selected segments. The corpus contains recordings of different speakers *and at different amplitudes*. We do *not* normalise the utterances to equal levels – thus allowing for combinations of speakers where one speaker can be up to 12dB louder than the other.

The ECAPA-TDNN is trained on the Voxceleb 1&2 database [29], where audio of around 7250 celebrities, and in different environmental settings, is scraped from YouTube. Thus, it is trained on *completely different* data than that used in the evaluation.

5.6.4 Parameter settings

All audio signals are sampled at 16kHz. A von Hann window of length 512 samples (32ms) and window shift of 160 samples (10ms) is applied before computing the STFT representations. The MFCC parameters are: $L = 16$ and $Q = 8$. Discarding the zeroth MFCC-bin, we take the first $\mathcal{N} = 13$ elements, resulting in a 39-dimensional feature vector $\boldsymbol{\mathcal{F}}^{\text{MFCC}}$.

The speaker verification feature $\mathcal{F}^{\text{SpVer}}$ length is 192. However, we note that a longer feature vector does not necessarily lead to more informative features for the Mod-MFCC feature representation. The averaging factor B for the mask computation in (5.14) and (5.19) is set to 5. For clustering, we use the fuzzy C-means python package [30].

5.6.5 Evaluation metrics

Defining good performance metrics that can quantify the clustering quality is not straightforward as it is difficult to define a ground truth.

Attempts have been made to generate ground truths with the help of oracle knowledge of either microphone-source distances [4, 6] or the RIRs [3]. However, the former fails to convey the full picture regarding the signal mixing (it considers strictly circular boundaries without, e.g., accounting for the sound propagation along indirect paths). Using the oracle RIRs for the ground truths does solve the problem of creating non-circular boundaries, but is not easily adaptable to include background clusters or variations in signal levels.

Generating such ground truths also has the disadvantage of forcing hard cut-offs – which does not well-describe the soft transition between clusters. The normalised cluster-centroid–to–source distance metric used in [9] does give more informative results in that sense. However, it also does not convey the full picture of the signal mixture (*e.g.*if one source speaks louder than the other one), and thus assumes that a circular distribution around the target speaker is the ideal result.

Therefore, we proposed 3 additional metrics in [13], which should provide an intuitive means of quantifying the clustering. This is briefly discussed in Section 5.6.5.1. We also note that an indirect way to evaluate the cluster quality is by evaluating the performance of the subsequent tasks, e.g., [2] evaluates the performance based on the results of a gender classification task. In this paper, we evaluate the clusters based on standard instrumental metrics for speaker separation, which will be explained in Section 5.6.5.2.

5.6.5.1 Metrics to evaluate clustering quality

The goal of our 3 alternative metrics is to allow an intuitive interpretation of the clustering performance. Since the underlying aim is source separation, a clustering that favours microphones with a strong direct-path component and a good signal to interference and noise ratio would be desirable. Accordingly, we compute (i) the direct-to-reverberant ratio (DRR) and (ii) the direct-to-reverberant, interference, and noise ratio (DRINR) for each microphone m allocated to a *speech-source* cluster. To this end, we split source signal $x_{j,m}(n)$ into the direct path component

$x^{\text{dir}}_{j,m}$ and the reflections $x^{\text{rev}}_{j,m}$:

$$x_{j,m}(n) = x^{\text{dir}}_{j,m}(n) + x^{\text{rev}}_{j,m}(n). \tag{5.20}$$

Then the DRR and DRINR are defined as follows:

$$\text{DRR} = \frac{\sum_n (x^{\text{dir}}_{c,m}(n))^2}{\sum_n (x^{\text{rev}}_{c,m}(n))^2} \quad \text{and} \tag{5.21}$$

$$\text{DRINR} = \frac{\sum_n (x^{\text{dir}}_{c,m}(n))^2}{\sum_n (y_m(n) - x^{\text{dir}}_{c,m}(n))^2} \tag{5.22}$$

Subsequently, we plot the *distribution* of these values. A distribution centred around high DRRs and DRINRs values indicates that the clustering selects only those microphones with relevant information about the speaker of that cluster. Lastly, the third metric indicates the amount of spatial diversity available from the clustering. This is computed as the average number of microphones allocated to a speech cluster.

5.6.5.2 Source separation metrics

As previously noted, clustering quality is indirectly reflected by performance in the subsequent tasks. Here, we use source separation metrics for this purpose, under the reasonable assumption that good clusters would lead to good source separation. We consider 3 standard and widely used instrumental metrics for source separation: the first is the source-to-interference ratio (SIR), as defined by [31]. This is an important metric for the initial masks since the masked signals are used to estimate the TDOA for subsequent delay compensation in the DSBs. After applying the initial masks, it is crucial that only the target source is present for a correct TDOA estimation.

However, interference and noise suppression is only a part of the story. We also use the short-time objective intelligibility (STOI) [32] and the perceptual evaluation of speech quality (PESQ) [33]) metrics to quantify the target-attenuation.

5.7 Results and discussion

5.7.1 First set of scenarios - sources far apart

The results are plotted in Figures 5.3 to 5.5 and Table 5.1. We first take a look at the results for the Euclidean distance only, since that corresponds with the results for the shoe-box acoustics presented in [13]. Here, we see fairly similar patterns: in Figure 5.3, the notched box-plots show that the speaker verification features lead to better speaker separation metrics, with statistical significance at the median level. This is true for all separation methods (x-axis) and evaluation

metrics (subfigures). Figures 5.4a and 5.5a also show that the Mod-MFCC features tend to include more microphones with relatively low source dominance (low DRR and DRINR). In contrast, the histogram plots for the speaker embeddings are narrower and include a larger number of microphones with relatively high DRRs and DRINRs, suggesting that using the speaker embeddings allows the clustering to find more useful microphones. Additionally, Table 5.1 also indicates that the cluster size when using speaker embeddings is larger than that using mod-MFCC features. This combination of a larger number of microphones which have, on average, better source dominance (high DRR and DRINR), indeed makes it possible to improve separation - which is seen in the separation metrics. For comparison, the metrics computed on the reference microphone for each cluster is also plotted (first column).

Interestingly, when using the cosine distance metric, the performance of the Mod-MFCC features improves greatly and the separation performance becomes comparable to the separation performance when using speaker embedding based features. The improvement is less evident for the speaker-embedding based features. The DRR and DRINR distributions in Figures 5.4b and 5.5b indicate, the speaker embeddings in combination with the cosine distance yield ever so slightly narrower histograms compared to using the Euclidean distance. This may be verified more straightforwardly from the DRINR histograms for Euclidean and cosine distance in Figure 5.6. Thus, the tendency is towards the selection of fewer lower-quality microphones for each source cluster. This improved microphone selection translates to, similarly, a slightly better separation performance, visible in Figure 5.3, where there are fewer outliers and a more compact boxplot.

When comparing the average number of microphones per source cluster (Table 5.1), there are fewer microphones on average when using the cosine distance. However, the DRR and DRINR distributions indicate that the microphones that are omitted are mostly of lower quality.

In general, we can conclude that for this set of scenarios, the cosine distance metric is better than the Euclidean distance. The improvement is most marked for Mod-MFCC based features. The combination of cosine distance and Mod-MFCC based features yields clusters with *separation* performance comparable to that using speaker embedding based features. Also, we obtain the same trends in

Table 5.1: Average number of microphones per source *cluster and choice of distance metric. Results for the first set of scenarios, where the sources are placed relatively far from each other. The larger the number of microphones, the more the spatial diversity available for a cluster.*

	Euclidean	cosine
MFCC	4.64	4.57
SpVer	4.84	4.76

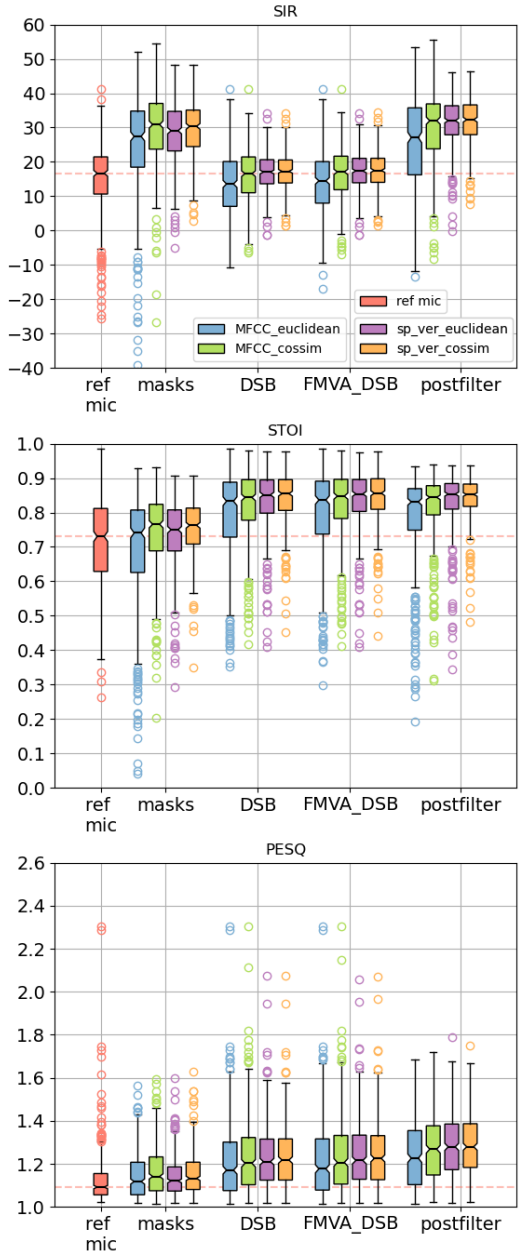
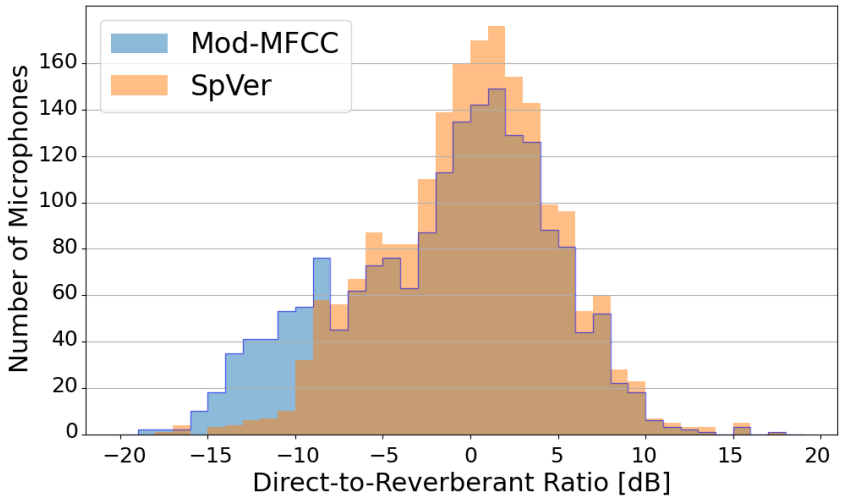
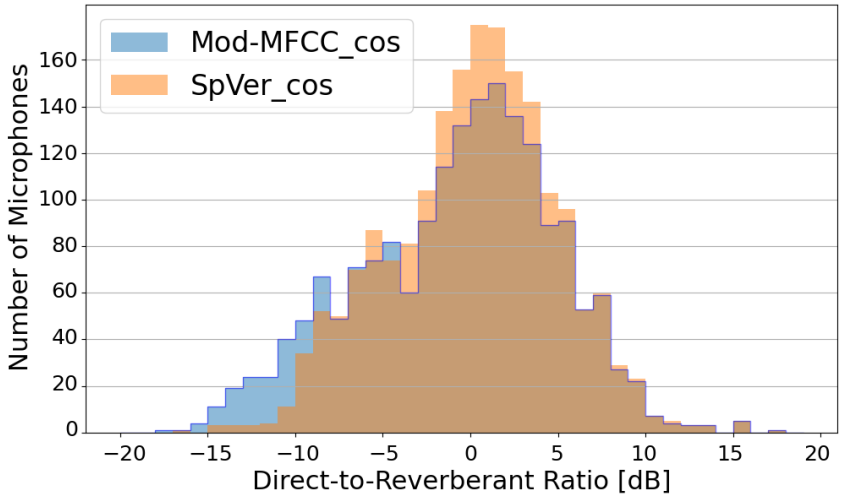


Figure 5.3: Performance metrics (SIR, PESQ, STOI) showing the separation effectiveness of the cluster feature types (colours) and method (x-axis) for the first set of scenarios, where the sources are always sufficiently far apart. For this and the other scenarios, some audio examples are available at https://aspire.ugent.be/demos/EURASIP2023SK/ .

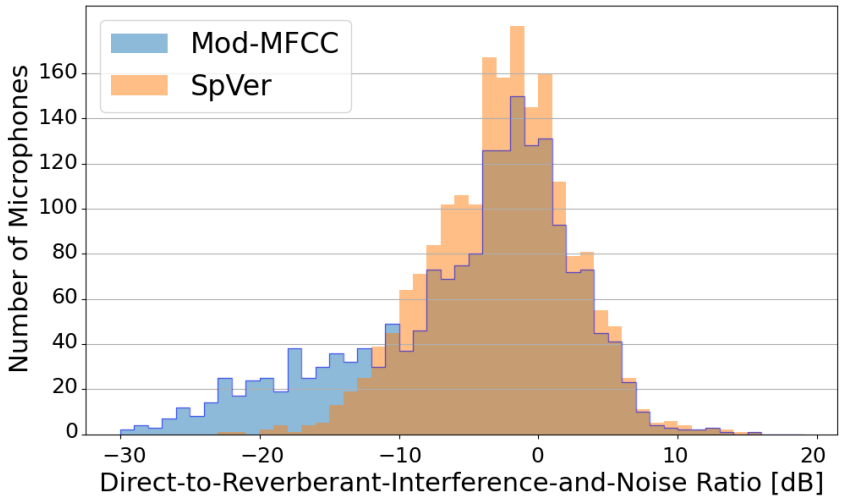


(a) Euclidean distance

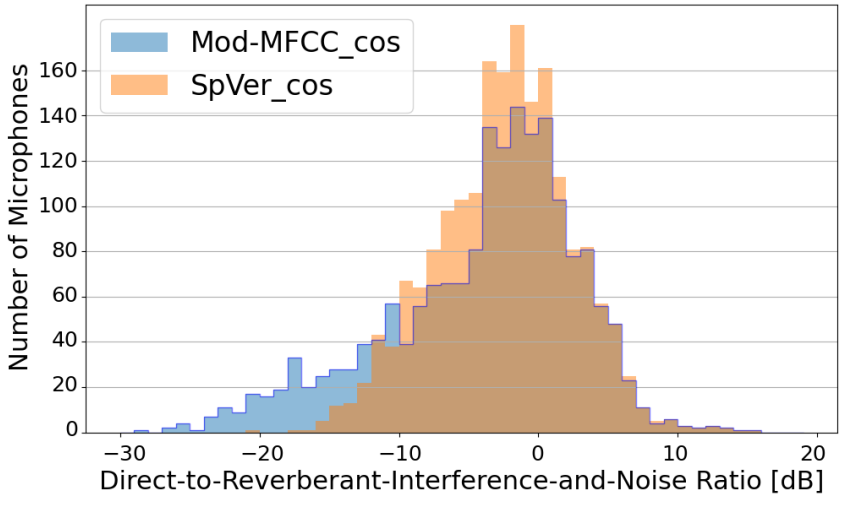


(b) Cosine distance

Figure 5.4: DRR histograms with (a) the Euclidean distance or (b) the cosine distance for the first set of scenarios. In this set, the sources are located quite far apart. The DRRs are computed only for microphones that are part of a source cluster. SpVer indicates the clusters based on the speaker verification features, while Mod-MFCC indicates the results of the clusters based on the Mod-MFCC features.



(a) Euclidean distance



(b) Cosine distance

Figure 5.5: DRINR histograms with (a) the Euclidean distance or (b) the cosine distance for the first set of scenarios. In this set, the sources are located quite far apart. The DRINRs are computed only for microphones that are part of a source cluster. SpVer indicates the clusters based on the speaker verification features, while Mod-MFCC indicates the results of the clusters based on the Mod-MFCC features.

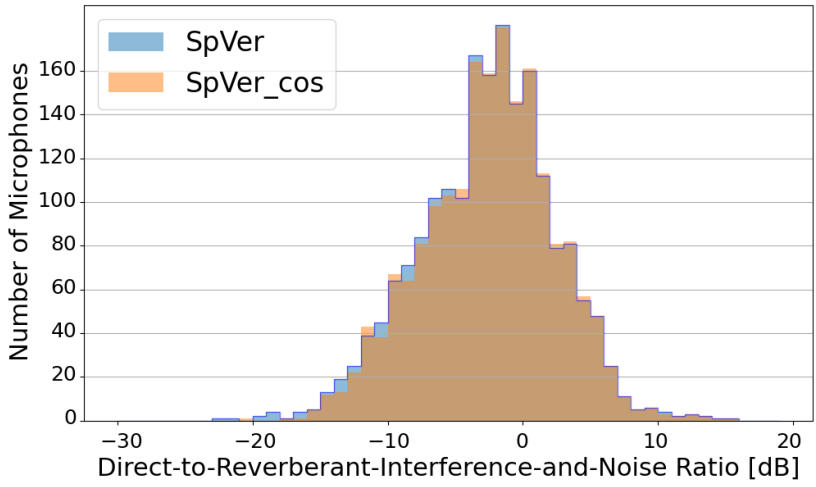


Figure 5.6: DRINR histograms of the first set of scenarios (sources are far) for the speaker embedding (speaker verification features) based clustering. The DRINRs are computed only for microphones that are part of a source cluster.

performance in this realistic setting as we obtained using the simulated (shoe-box acoustics) rooms.

5.7.2 Second set of scenarios - sources in close proximity

The results for this more challenging setting are presented in Figures 5.7 to 5.9 and Table 5.2. Bringing the sources closer, unsurprisingly, makes the clustering harder. While the separation performance using speaker embedding features still outperforms the mod-MFCC-based features, all speaker separation metrics are lower than for the first scenario. SIRs are even, sometimes, below $0\,dB$. However, since the sources can have different signal amplitudes, it is possible that for such close sources, one source dominates, making it nearly impossible to separate those with the chosen simple separation scheme. This highlights the importance of more sophisticated separation approaches.

One interesting observation on the speaker separation metrics is that the cosine distance seems to give a significant improvement over the Euclidean distance, and for *both* sets of features. This is most prominent for the initial mask estimate, which indicates that using the cosine distance yields a better *reference microphone* for each cluster.

The histograms in Figures 5.8 and 5.9 tell a similar story, where the speaker embedding features select more useful microphones. Nevertheless, it is instructive to zoom in on the region of less than ideal microphones (-10dB DRR and lower) in Figure 5.8a. There, the DRR histogram would suggest that the Mod-MFCC fea-

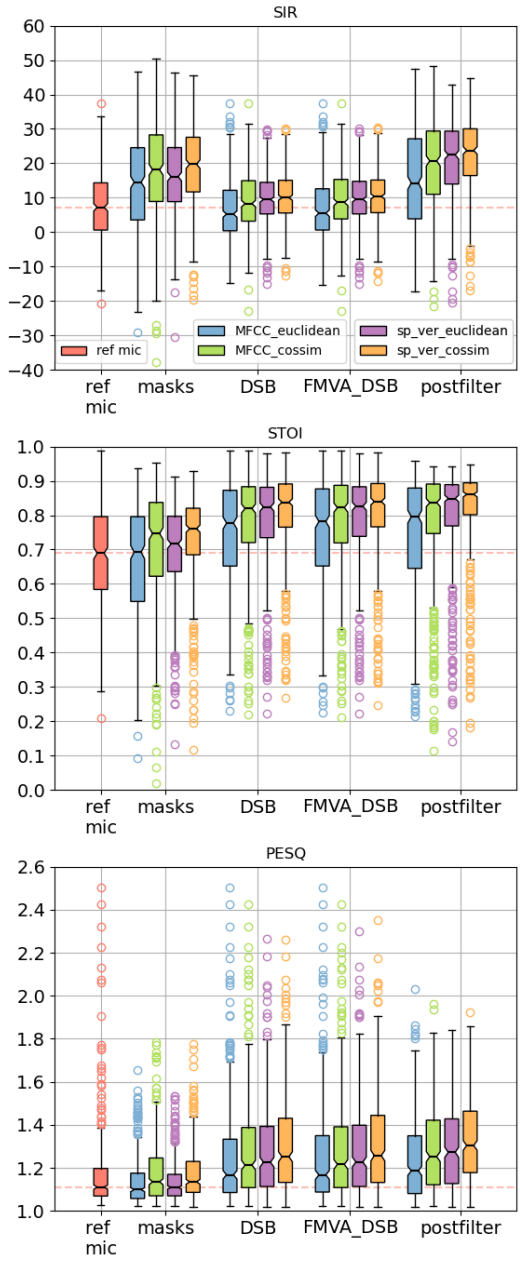
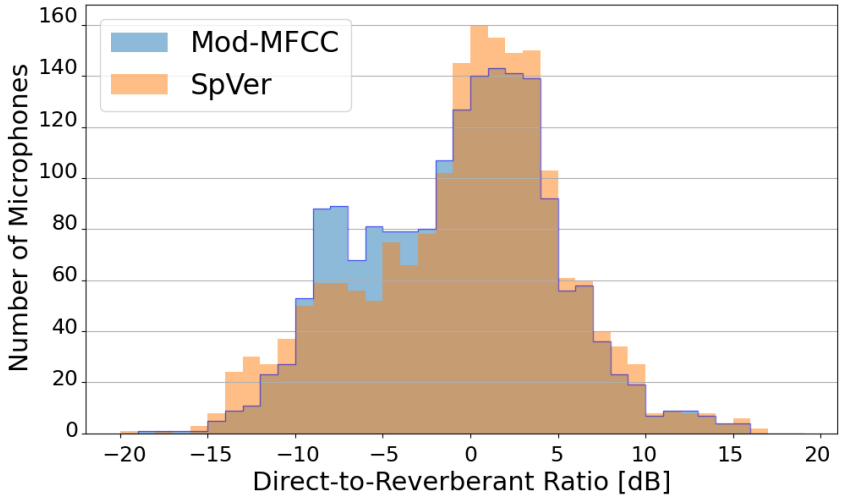
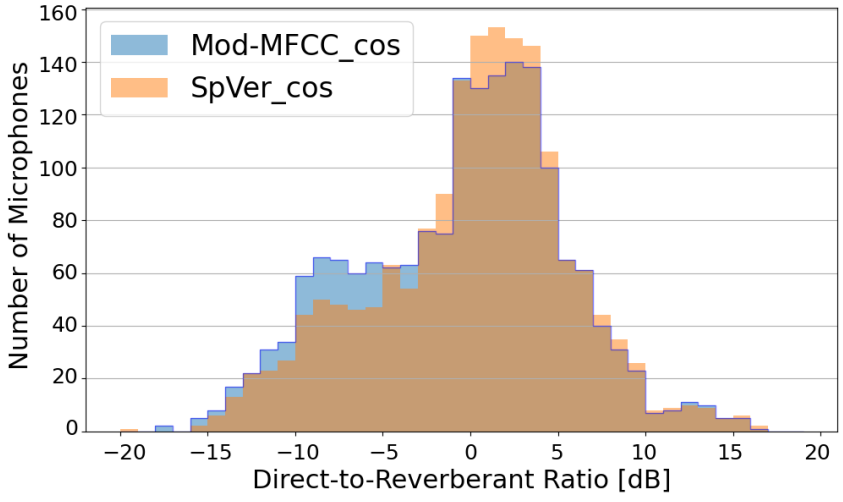


Figure 5.7: Performance metrics (SIR, PESQ, STOI) showing the separation effectiveness of the cluster feature types (colours) and method (x-axis) for the second set of scenarios, where the sources are maximally separated by three times the critical distance.

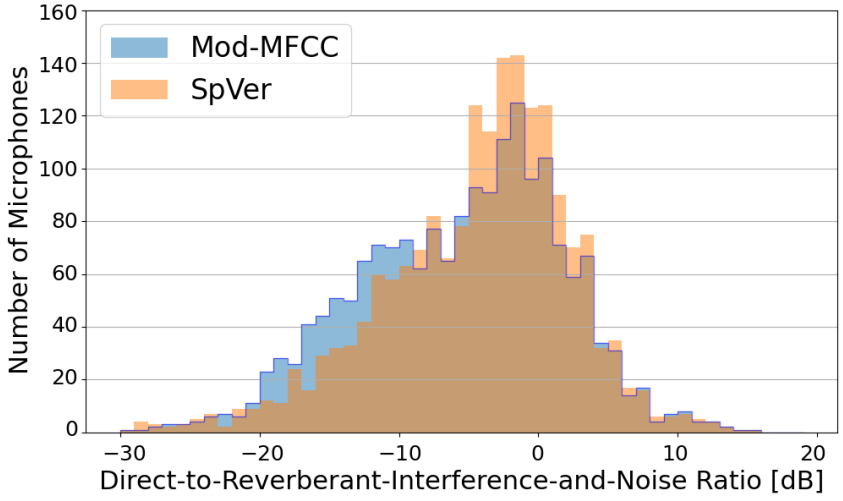


(a) Euclidean distance

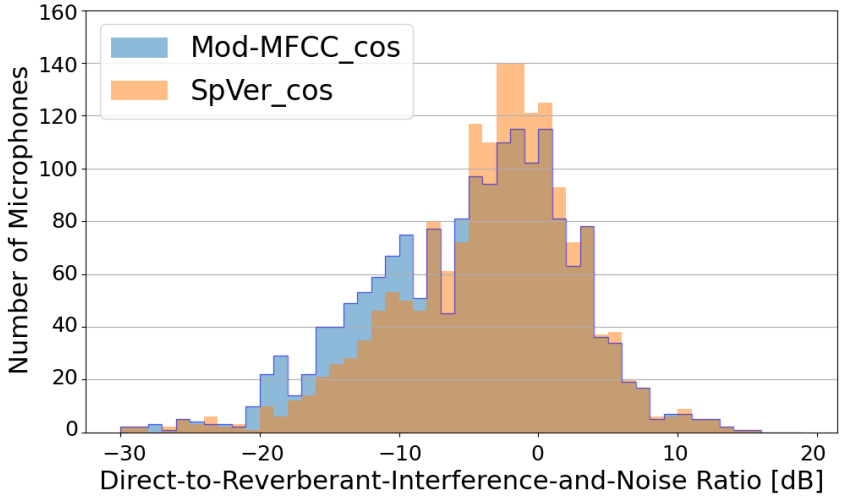


(b) Cosine distance

Figure 5.8: Histograms of DRR with (a) the Euclidean distance or (b) the cosine distance for the second set of scenarios. In this set, the sources are separated by at most three time the critical distance. The DRRs are computed only for microphones that are part of a source cluster.



(a) Euclidean distance



(b) Cosine distance

Figure 5.9: Histograms of the DRINR with (a) the Euclidean distance or (b) the cosine distance for the second set of scenarios. In this set, the sources are separated by at most three time the critical distance. The DRINRs are computed only for microphones that are part of a source cluster.

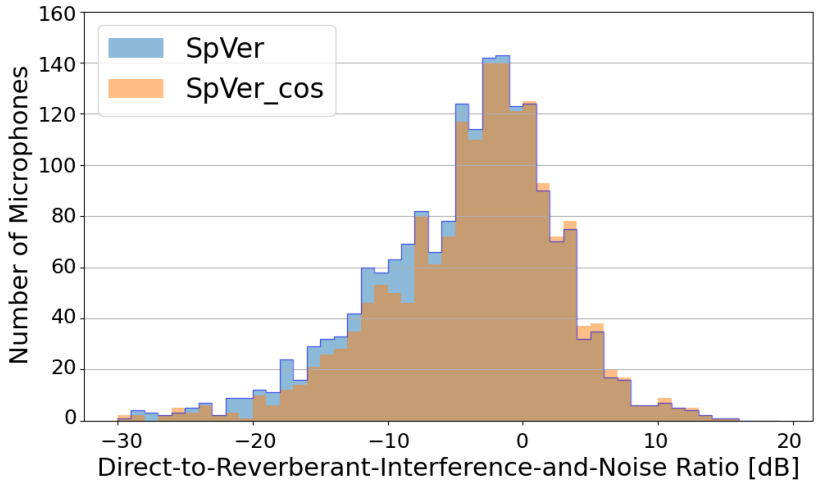


Figure 5.10: DRINR histograms for the second set of scenarios (sources are close) for the speaker embedding based clustering. The DRINRs are computed only for microphones that are part of a source cluster.

tures lead to a better microphone allocation than the speaker embeddings. However, when looking at the DRINR distribution in Figure 5.9a, the conclusions seem to be reversed. This indicates that the speaker embeddings are better at incorporating information about the target and interference speaker for the clustering, rather than only the distance of a microphone to the target speaker (which is likely what the Mod-MFCC based features focus on). Note that this is mainly for the Euclidean distance metric. The results are more consistent when using cosine similarity. Additionally, Figure 5.10 does demonstrate a clear benefit of the cosine distance in combination with speaker embeddings, making the cosine distance more beneficial in situations where the sources are close compared to situations where the sources are distributed further apart in the room (Figure 5.6).

Table 5.2 shows that for the Euclidean distance, a similar conclusion as for the previous sections is applicable: speaker verification features generate slightly larger clusters, and of higher quality (seen from the DRR and DRINR histograms). For the cosine distance, the number of microphones does not significantly change

Table 5.2: Average number of microphones per source *cluster and choice distance metric for the second set of scenarios, where the sources are placed in close proximity to each other.*

	Euclidean	cosine
MFCC	4.54	4.39
SpVer	4.64	4.34

between the choice of features. Again, the number of microphones decreases when using the cosine distance, but it is mainly the lower quality microphones that are removed (conclusion from the DRINR plots in Figure 5.9).

5.7.3 Effect of segment length

Figures 5.11 to 5.13 and Table 5.3 show the impact of shortening the length of the segment of the signal given to the feature extractors. The experiments were carried out for the same set of scenarios as Section 5.7.1 and using only the cosine similarity, since it yielded the best results in the previous experiments.

For the Mod-MFCC features, the clusters consistently degrade as the segment lengths decrease and more drastically for lengths of 1s and 0.5s. In contrast, for the speaker embedding features, the segment length seems to have only a marginal impact on the clustering capability, even for the short length of 0.5s. This is further visible in both the DRR and DRINR distributions, where those for the speaker embeddings have only a very slight shift towards lower DRRs and DRINRs, while for the MFCC-based features, the shift is marked, becoming increasingly prominent for shorter evaluation lengths.

The same effect is visible in the speaker separation metrics in Figure 5.11: the performance of the Mod-MFCC based features again starts dropping with lower segment lengths. In contrast, the performance of the embedding based speaker separation stays quite consistent.

In terms of the average number of microphones per cluster – this does not drastically change for different evaluation lengths for the speaker embeddings, while for the Mod-MFCC, the number of microphones *increase*. This larger cluster mainly contains microphones with a poor signal to interference-and-noise ratio (visible in the DRINR histogram Figure 5.13a), which negatively impacts the separation performance.

This leads us to the satisfactory conclusion that speaker embeddings computed on short segments still yield robust features for clustering *ad hoc* distributed microphones. This can be utilised to lower the computational complexity since the feature extraction has shorter input segments. Alternatively, the robustness of the features to shorter segment lengths can be exploited to quickly adapt the clustering in more dynamic scenarios.

Table 5.3: Average number of microphones per source *cluster for different evaluation lengths. The cosine distance metric is used throughout.*

	4s	2s	1s	0.5s
MFCC	4.52	4.65	5.50	5.52
SpVer	4.71	4.68	4.77	4.90

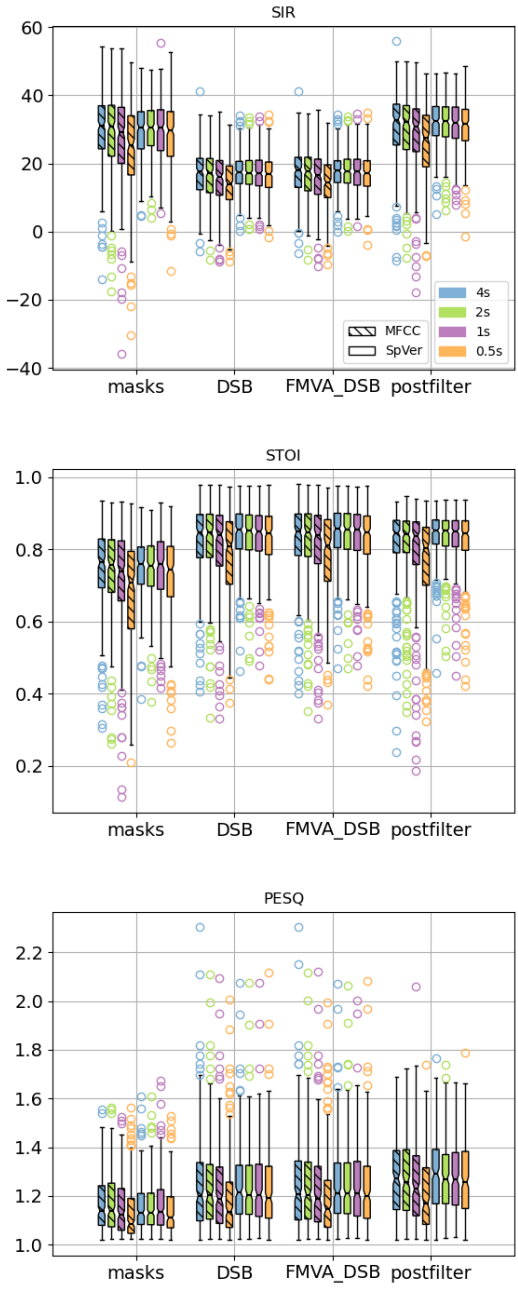
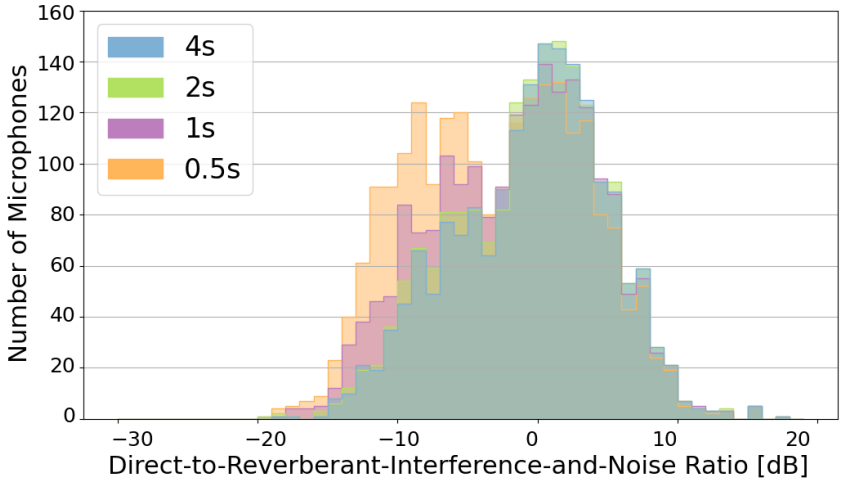
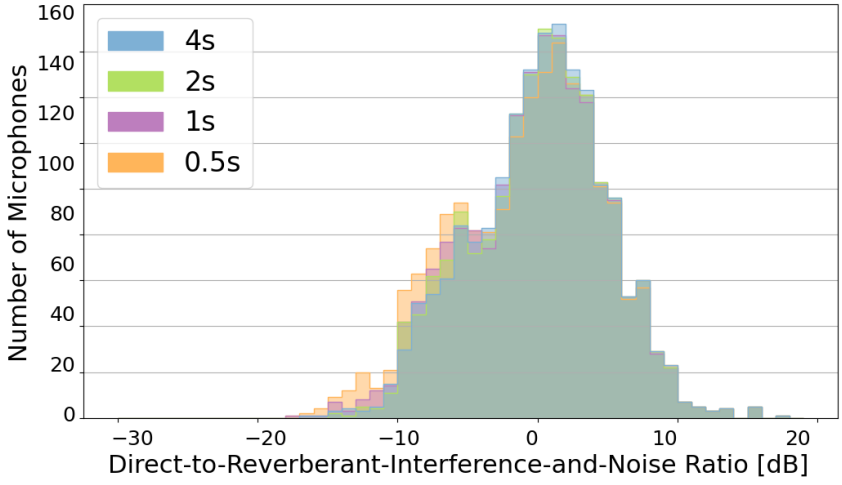


Figure 5.11: Performance metrics (SIR, PESQ, STOI) showing the separation effectiveness of the cluster feature types (hatches), method (x-axis) and duration (colour) for the first set of scenarios, where the sources are always sufficiently far apart.

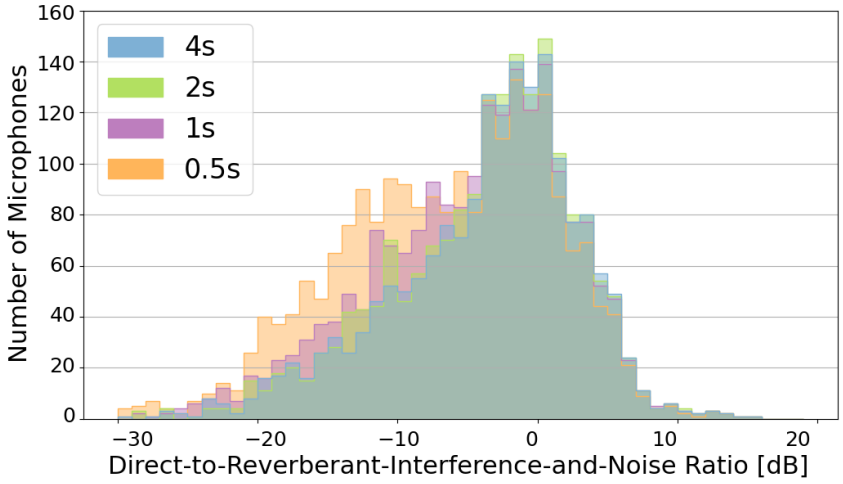


(a) MFCC

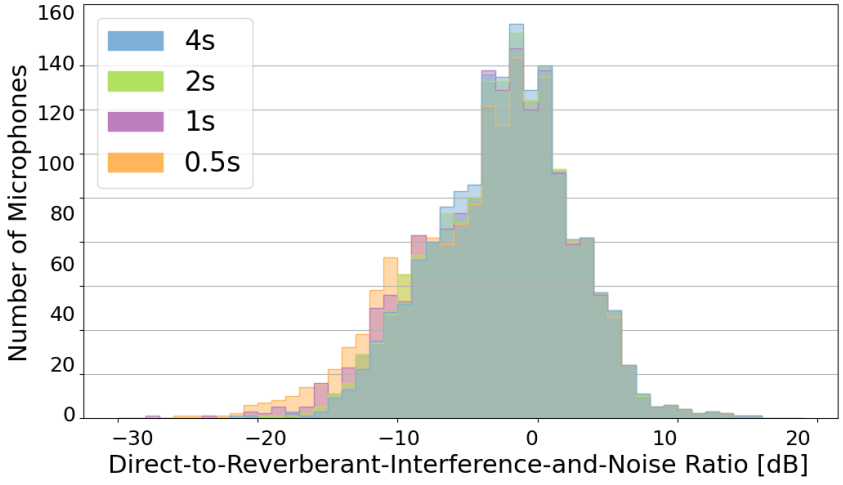


(b) SpVer

Figure 5.12: Histograms of the DRRs of the (a) Mod-MFCC features (b) and speaker verification features for different evaluation durations and the cosine distance metric. These are computed only for microphones that are part of a source cluster.



(a) MFCC



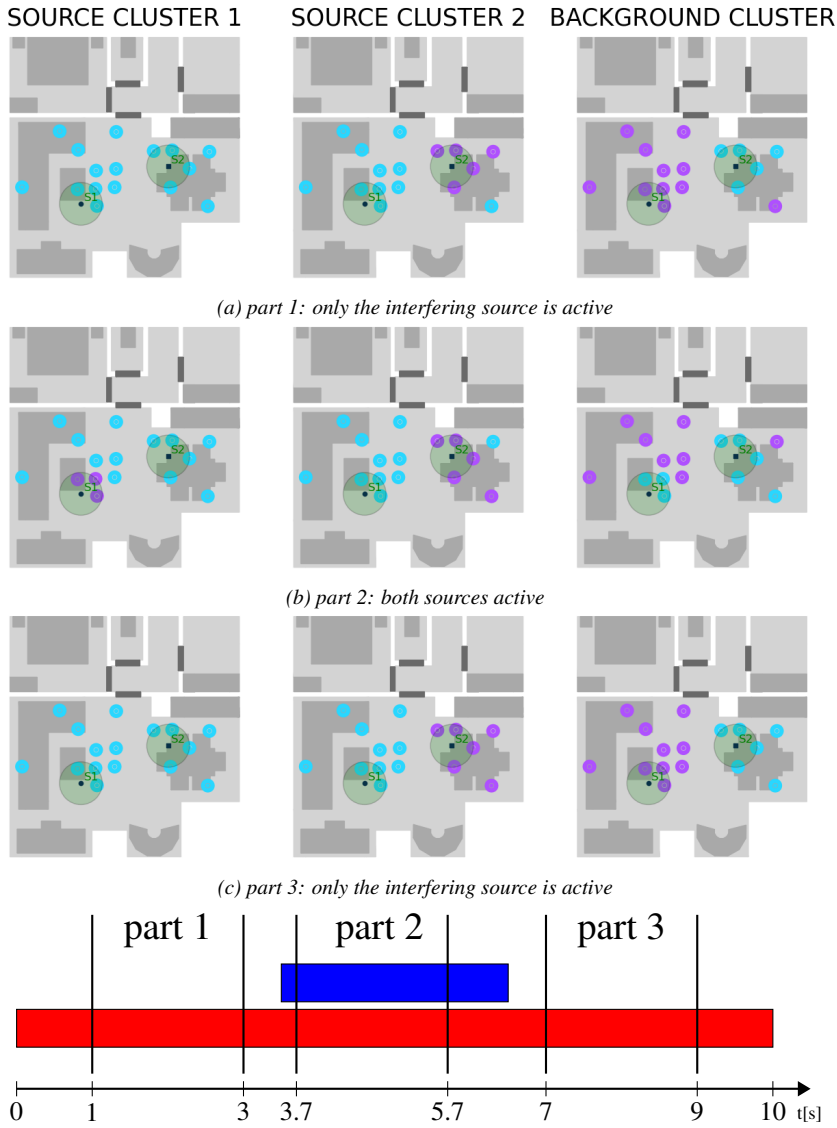
(b) SpVer

Figure 5.13: Histograms of the DRINRs of the (a) Mod-MFCC features (b) and speaker verification features for different evaluation durations and the cosine distance metric. These are computed only for microphones that are part of a source cluster.

5.7.4 Known speaker embedding

Having shown the general robustness of the speaker embedding features, applied to the task of clustering *ad hoc* distributed microphones, we focus on investigating whether these features can be exploited to focus on only a *desired* subset of speakers, which are known *a priori*. We only show one example for this scenario - more as an empirical proof-of-concept. Since the scenario is dynamic, we use shorter evaluation segments of length $2\,s$. Figure 5.14 shows the results for this scenario.

In the first and last part (Figures 5.14a and 5.14c), where only the interferer is active, there is an empty source cluster for the target source. The FCM generates two other cluster centres that model the interfering source and the background characteristics. Note that in these periods it is desirable that the microphones close to the inactive speaker are grouped in the background cluster. During the period when the target source is active, the features extracted from microphones close to the source match the known-speaker embedding (which is used to initialise the cluster centre for this source) and the FCM faithfully attributes the appropriate microphones to this source – as can be seen in Figure 5.14b. This demonstrates, in addition to robustness, the ability to induce microphone clustering in a speaker selective manner. This constitutes an additional benefit of using embeddings as clustering features. Such behaviour would not be straightforward to implement using other features, *e.g.*, those purely based on room characteristics.



(a) part 1: only the interfering source is active

(b) part 2: both sources active

(c) part 3: only the interfering source is active

Figure 5.14: Incorporation of known speaker embedding for targeted clustering. In this scenario, the interfering source is active throughout the experiment. However, the target (known) source is only active for a short period in the middle of the segment. This is schematically indicated above, where the blue bar indicates the time-period where the known speaker is active, while red indicates the interferer activity. We initialise the first cluster with the known speaker embedding. Thereby, the FCM algorithm generates an empty cluster in parts (a) and (c) and only allocates microphones to the target cluster when the known source is active. In the figure, a microphone is part of the cluster if its colour is dark purple, while the light blue colour indicates that it is not part of the cluster.

5.8 Conclusions

Our prior work, which introduced speaker embeddings as robust features for clustering *ad hoc* distributed microphones, raised several interesting follow-up questions that were addressed in this paper. Firstly, we evaluated the performance of speaker embedding features in realistic settings and demonstrated similar trends as previously reported using simulations based on shoe-box acoustics models. Next, the effect of the distance metric used in the clustering algorithm was investigated and it was shown that the cosine distance offers more discriminative clustering compared to the Euclidean metric used previously. The benefit of this metric was more marked for the baseline mod-MFCC features, bringing their performance to a level comparable to that of the speaker embedding features, in scenarios where the sources are far apart. In more challenging conditions, however, the speaker embedding features, in combination with the cosine distance metric, better exploit the source-specific information and significantly outperform the Mod-MFCC based features.

In view of practical implementations, the effect of shorter segment lengths on the clustering performance was studied. Here, whereas mod-MFCC-based features consistently degrade with shorter segment lengths, speaker embedding features are only marginally affected and their performance remains more-or-less constant. Even with a segment length of 0.5 seconds, the clusters stay similar to the baseline. This robustness of the speaker embedding features can be exploited for two purposes: complexity reduction and/or quicker adaptation in dynamic scenarios. Complexity can be scaled by only computing the features and updating the clusters sporadically and using only a small amount of data, sampled over a wider time-range. To allow for quick adaptation in more dynamic scenarios, the idea would be to similarly compute the features over short, but contiguous time-intervals and update the clusters more frequently.

Lastly, we presented a proof-of-concept of how speaker embeddings could be used to explicitly incorporate information on a known speaker for targeted clustering and separation. In future work, we aim to further focus on this setting and incorporate not only more sophisticated separation approaches, like MVDR beamforming [22, 23, 34] or deep learning based methods [35], but also the improved embedding extractor proposed in [36], which offers increased robustness of the extracted embedding in the presence of interfering speech. Additionally, a comparison with spatially based cluster algorithms, like [3], should be performed, to see the trade-offs between the methods. Investigating an optimal combination of spatial and speaker-specific information is also an interesting path we shall explore in future work.

Extra clustering examples and the associated audio corresponding to the presented work are available at https://aspire.ugent.be/demos/EURASIP2023SK/.

Acknowledgment

This work is supported by the Research Foundation - Flanders (FWO) under grant number G081420N and imec.ICON: BLE2AV (support from VLAIO). Partners: Imec, Televic, Cochlear, and Qorvo.

References

[1] A. Bertrand. *Applications and trends in wireless acoustic sensor networks: A signal processing perspective*. In 2011 18th IEEE symposium on communications and vehicular technology in the Benelux (SCVT), pages 1–6. IEEE, 2011.

[2] S. Gergen, A. Nagathil, and R. Martin. *Classification of reverberant audio signals using clustered ad hoc distributed microphones*. Signal Processing, 107:21–32, 2015.

[3] A. J. Muñoz-Montoro, P. Vera-Candeas, and M. G. Christensen. *A Coherence-based Clustering Method for Multichannel Speech Enhancement in Wireless Acoustic Sensor Networks*. In 2021 29th European Signal Processing Conference (EUSIPCO), pages 1130–1134. IEEE, 2021.

[4] I. Himawan, I. McCowan, and S. Sridharan. *Clustered blind beamforming from ad-hoc microphone arrays*. IEEE Transactions on Audio, Speech, and Language Processing, 19(4):661–676, 2010.

[5] S. Gergen, R. Martin, and N. Madhu. *Source separation by fuzzy-membership value aware beamforming and masking in ad hoc arrays*. In Speech Communication; 13th ITG-Symposium, pages 1–5. VDE, 2018.

[6] S. Pasha, Y. X. Zou, and C. Ritz. *Forming ad-hoc microphone arrays through clustering of acoustic room impulse responses*. In 2015 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP), pages 84–88. IEEE, 2015.

[7] Y. Zhao, J. K. Nielsen, J. Chen, and M. G. Christensen. *Model-based distributed node clustering and multi-speaker speech presence probability estimation in wireless acoustic sensor networks*. The Journal of the Acoustical Society of America, 147(6):4189–4201, 2020.

[8] M. Dziubany, R. Machhamer, H. Laux, A. Schmeink, K.-U. Gollmer, G. Burger, and G. Dartmann. *Machine Learning Based Indoor Localization Using a Representative k-Nearest-Neighbor Classifier on a Low-Cost IoT-Hardware*. In 2018 26th European Signal Processing Conference (EUSIPCO), pages 2050–2054, 2018. doi:10.23919/EUSIPCO.2018.8553155.

[9] S. Gergen and R. Martin. *Estimating source dominated microphone clusters in ad-hoc microphone arrays by fuzzy clustering in the feature space*. In Speech Communication; 12. ITG Symposium, pages 1–5. VDE, 2016.

[10] S. Gergen, R. Martin, and N. Madhu. *Source separation by feature-based clustering of microphones in ad hoc arrays*. In 2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC), pages 530–534. IEEE, 2018.

[11] A. Nelus, R. Glitza, and R. Martin. *Estimation of microphone clusters in acoustic sensor networks using unsupervised federated learning*. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 761–765. IEEE, 2021.

[12] L. Becker, A. Nelus, R. Glitza, and R. Martin. *Accelerated Unsupervised Clustering in Acoustic Sensor Networks Using Federated Learning and a Variational Autoencoder*. In 2022 International Workshop on Acoustic Signal Enhancement (IWAENC), pages 1–5. IEEE, 2022.

[13] S. Kindt, J. Thienpondt, and N. Madhu. *Exploiting Speaker Embeddings for Improved Microphone Clustering and Speech Separation in ad-hoc Microphone Arrays*. In ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1–5. IEEE, 2023.

[14] B. Desplanques, J. Thienpondt, and K. Demuynck. *ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN based speaker verification*. In Interspeech 2020, pages 3830–3834. International Speech Communication Association (ISCA), 2020.

[15] R. Glitza, L. Becker, A. Nelus, and R. Martin. *Database of simulated room impulse responses for acoustic sensor networks deployed in complex multi-source acoustic environments*. In 2023 31st European Signal Processing Conference (EUSIPCO), pages 246–250. IEEE, 2023.

[16] F. Bimbot, J. F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-García, D. Petrovska-Delacrétaz, and D. A. Reynolds. *A tutorial on text-independent speaker verification*. EURASIP Journal on Advances in Signal Processing, 2004(4):1–22, 2004.

[17] P. N. Garner. *Cepstral normalisation and the signal to noise ratio spectrum in automatic speech recognition*. Speech Communication, 53(8):991–1001, 2011.

[18] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur. *X-Vectors: Robust DNN Embeddings for Speaker Recognition*. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5329–5333, 2018. doi:10.1109/ICASSP.2018.8461375.

[19] J. Hu, L. Shen, and G. Sun. *Squeeze-and-Excitation Networks*. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7132–7141, 2018. doi:10.1109/CVPR.2018.00745.

[20] J. Deng, J. Guo, N. Xue, and S. Zafeiriou. *ArcFace: Additive Angular Margin Loss for Deep Face Recognition*. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 4685–4694, 2019. doi:10.1109/CVPR.2019.00482.

[21] J. C. Bezdek, R. Ehrlich, and W. Full. *FCM: The fuzzy c-means clustering algorithm.* Computers & geosciences, 10(2-3):191–203, 1984.

[22] S. Markovich-Golan, A. Bertrand, M. Moonen, and S. Gannot. *Optimal distributed minimum-variance beamforming approaches for speech enhancement in wireless acoustic sensor networks*. Signal Processing, 107:4–20, 2015.

[23] D. Cherkassky, S. Markovich-Golan, and S. Gannot. *Performance analysis of MVDR beamformer in WASN with sampling rate offsets and blind synchronization*. In 2015 23rd European Signal Processing Conference (EUSIPCO), pages 245–249. IEEE, 2015.

[24] S. Rickard and O. Yilmaz. *On the approximate W-disjoint orthogonality of speech*. In 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing, volume 1, pages I–529. IEEE, 2002.

[25] G. Dekkers, S. Lauwereins, B. Thoen, M. W. Adhana, H. Brouckxon, B. Van den Bergh, T. Van Waterschoot, B. Vanrumste, M. Verhelst, and P. Karsmakers. *The SINS database for detection of daily activities in a home environment using an acoustic sensor network*. Detection and Classification of Acoustic Scenes and Events 2017, pages 1–5, 2017.

[26] B.-I. Dalenbäck. *TUCT v2.0e:1, CATT*. http://www.catt.se, 1999. Accessed: 2019.

[27] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. *Librispeech: an asr corpus based on public domain audio books*. In 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP), pages 5206–5210. IEEE, 2015.

[28] A. Nelus, R. Glitza, and R. Martin. *Unsupervised Clustered Federated Learning in Complex Multi-source Acoustic Environments*. In 2021 29th European Signal Processing Conference (EUSIPCO), pages 1115–1119. IEEE, 2021.

[29] J. S. Chung, A. Nagrani, and A. Zisserman. *VoxCeleb2: Deep Speaker Recognition*. Proc. Interspeech 2018, pages 1086–1090, 2018.

[30] M. L. D. Dias. *Fuzzy C-means: An implementation of Fuzzy C-means clustering algorithm.*,". https://git.io/fuzzy-c-means, May 2019. Available from: https://git.io/fuzzy-c-means. Available from: https://git.io/fuzzy-c-means, doi:10.5281/zenodo.3066222.

[31] E. Vincent, R. Gribonval, and C. Févotte. *Performance measurement in blind audio source separation*. IEEE transactions on audio, speech, and language processing, 14(4):1462–1469, 2006.

[32] 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, pages 4214–4217. IEEE, 2010.

[33] 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., volume 2, pages 749–752. IEEE, 2001.

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

[35] Z.-Q. Wang, J. Le Roux, and J. R. Hershey. *Multi-channel deep clustering: Discriminative spectral and spatial embeddings for speaker-independent speech separation*. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1–5. IEEE, 2018.

[36] J. Thienpondt, N. Madhu, and K. Demuynck. *Margin-Mixup: A Method for Robust Speaker Verification in Multi-Speaker Audio*. In 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2023.

6

Ad Hoc Distributed Microphones Clustering: A Comparative Analysis on Using Coherence and Signal-Specific Features

This chapter compares speaker-specific feature based clustering methods of Chapter 5 to a coherence based clustering method for ad-hoc distributed microphone settings. To our surprise, the coherence based method is slightly more effective than the feature based method, even with an audio codec that distorts the signal in a non-linear manner. Additionally, the reference microphone, which should ideally be the best microphone of each cluster, is also more robustly selected by the coherence based clustering. This improved selection is valuable, since in separation methods of Chapter 7, the reference microphone gets promoted with the assumption that it is indeed the best one.

⋆ ⋆ ⋆

Stijn Kindt*, Martijn Meeldijk*, Nilesh Madhu

Published in 15th ITG Conference on Speech Communication (ITG 2023).

* Equal contributions, shared first author

Abstract It is often useful to cluster ad hoc distributed microphones according to the dominant source each captures. For example, in a recently proposed source separation approach by Gergen *et al.*, inter- and intra-cluster information is aggregated to enhance the dominant source at each cluster. To generate the features for this blind clustering, spectro-temporal characteristics of the signals are usually exploited to be clustered by the fuzzy C-means algorithm. A recent alternative by Muñoz-Montoro *et al.*, uses the spatial relations between microphones, encoded by pairwise broadband coherence. Non-negative matrix factorisation of the positive, semi-definite coherence matrix thus obtained directly yields the cluster assignments. Here, we compare these two types of approaches in terms of the resultant cluster quality. However, coherence-based approaches require the transfer of each microphone signal to a central processing node for feature computation - in contrast to the feature-based approaches, which only require transmission of time-aggregated feature vectors. To counter this large bandwidth requirement of the coherence-based approach, we examine its performance under lossy encoding from a standard codec (BLE LC3plus). Results show that, contrary to intuition, the coherence-based approach remains robust to such non-linear encoding - making this a viable option for bandwidth-limited (wireless) acoustic sensor networks.

6.1 Introduction

Combining the information present in (wireless) acoustic sensor networks (WASNs), shows great potential to improve tasks such as speaker separation, diarization, and automatic speech recognition[1]. The advantage of utilising these WASNs over compact microphone arrays lies in their ability to provide additional spatial diversity. However, in the case the WASNs are constructed by ad hoc distributed microphones, their positions are not known a priori. By clustering microphones dominated by the same source, greater clarity can be achieved regarding the relative microphone positions, facilitating subsequent processing stages.

The clustering approaches discussed in [2] and [3] leverage room characteristics, utilising either diffuse noise characteristics or room impulse response (RIR) estimation. In contrast, alternative methods focus on extracting features for clustering. For instance, [4] introduced Modulation Mel Frequency Cepstral Coefficients (Mod-MFCC) as features. On the other hand, [5–7], and [8] propose the use of Deep Neural Networks (DNNs) to extract features, employing either an auto-encoder or a pre-trained speaker verification network.

Another approach exploits spatial features, specifically the magnitude squared coherence (MSC) computed between microphones [9]. These coherence values are then arranged in a coherence matrix, which is subsequently used to perform clustering via non-negative matrix factorisation (NMF) [10].

A comparison of the different techniques is currently missing. This paper aims

to address this gap by comparing feature-based methods such as Mod-MFCCs and speaker embeddings with coherence-based clustering. However, in its current form, the coherence-based method requires far more bandwidth, which is usually restricted in WASNs for energy and efficiency concerns [1]. Therefore, the coherence-based method will also be evaluated with the incorporation of an audio codec. The LC3plus codec [11], used in Bluetooth low-energy (BLE)applications, can reduce a signal to a bitrate of 16kbps. Although the bandwidth required for these signals still exceeds that of the feature-based methods, it should be noted that this allocation of bandwidth is not wasted, as subsequent processing would often require access to these signals.

The rest of the paper is structured as follows: the signal model is outlined in Section 6.2, after which the coherence-based clustering is detailed in Section 6.3. This is followed by an overview of the feature-based clustering methods in Section 6.4. The experiments and results are then presented in Section 6.5 and discussed in Section 6.6. Finally, in Section 6.7, the paper concludes by summarising the key findings and implications, as well as providing insights into potential future research directions.

6.2 Signal Model

The considered scenario consists of M microphones distributed in a room with J active sources. The signal at each microphone m is given by:

$$y_m(n) = \sum_{j=1}^{J} x_{j,m}^{\text{dir}}(n) + x_{j,m}^{\text{rev}}(n) \quad + v_m(n)\,, \tag{6.1}$$

with n the discrete time index. $x_{j,m}^{\text{dir}}$ is the direct path contribution of source signal from the jth source to the mth microphone, while $x_{j,m}^{\text{rev}}$ represents all the reflected components of the jth source. v_m is the additive noise at the microphone m. The short-time Fourier domain representation of the signal is represented by the corresponding capital letter:

$$Y_m(l,k) = \text{STFT}\big[y_m(n)\big]\,, \tag{6.2}$$

where l is the time index and k is the frequency bin. The Von Hann window is used to perform the STFT.

6.3 Coherence-Based Clustering

First, we discuss the previously proposed frequency domain coherence-based clustering method from [9]. This approach involves calculating the magnitude squared

coherence between microphone pairs and constructing a coherence matrix. Subsequently, non-negative matrix factorisation (NMF) is employed to cluster the microphones into C clusters.

6.3.1 Frequency-Domain Coherence

By utilising the magnitude squared coherence, it is possible to conduct an analysis of the linear relationship between two microphone signals $y_m(n)$ and $y_{m\prime}(n)$. The MSC is computed as:

$$\Gamma_{mm\prime}(k) = \frac{|P_{mm\prime}(k)|^2}{P_{mm}(k)P_{m\prime m\prime}(k)}\,, \tag{6.3}$$

where $P_{mm\prime}$ represents the cross power spectral density (PSD) between y_m and $y_{m\prime}$, while P_{mm} and $P_{m\prime m\prime}$ represents the auto Power spectral density of y_m and $y_{m\prime}$ respectively. The PSDs can be calculated in the STFT domain using methods such as Welch's method [12] or recursive averaging. A common approach is to perform averaging over 4-second sections [4]. The final coherence value is obtained by averaging over all frequency bins of the MSC:

$$\mathcal{C}_{mm\prime} = \frac{1}{K}\sum_{k=0}^{K-1}\Gamma_{mm\prime}(k) \in [0,1]\,. \tag{6.4}$$

Coherence values are computed for all microphone pairs and placed in $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{M\times M}$. Note that this matrix is non-negative and symmetrical and has the property $\boldsymbol{\mathcal{C}}_{mm} = 1$.

6.3.2 NMF-based Clustering

Non-Negative Matrix Factorisation (NMF) [13] is employed to calculate the cluster matrix $\boldsymbol{B} \in \mathbb{R}^{M\times C}$, where $\boldsymbol{B}_{mc}$ is the contribution of microphone m to cluster c. The symmetrical and diagonal properties of the coherence matrix $\boldsymbol{\mathcal{C}}$ can be exploited to write the problem as follows [9]:

$$\boldsymbol{\mathcal{C}} = \boldsymbol{B}\boldsymbol{B}^T \odot (\mathbf{1} - \boldsymbol{I}) + \boldsymbol{I}\,, \tag{6.5}$$

where $\odot$ denotes element-wise (Hadamard) product, $\boldsymbol{I}$ is the identity matrix, and $\mathbf{1}$ is the all-ones matrix.

It is possible to estimate $\boldsymbol{B}$ using iterative multiplicative update rules based on Euclidean divergence [13]:

$$\boldsymbol{B} \leftarrow \boldsymbol{B} \odot \frac{(\boldsymbol{\mathcal{C}} \odot (\mathbf{1} - \boldsymbol{I}))\boldsymbol{B}}{(\boldsymbol{B}\boldsymbol{B}^T \odot (\mathbf{1} - \boldsymbol{I}))\boldsymbol{B}}\,. \tag{6.6}$$

Due to the inherent clustering property of NMF [10], $\boldsymbol{B}$ consequently contains fuzzy membership values (FMVs) representing each microphone's contribution for each cluster. Note that these fuzzy values do not automatically sum up to one for each microphone. A normalisation step is carried out to ensure that: $\boldsymbol{B}_{mc} = \boldsymbol{B}_{mc} / \sum_{c=0}^{C-1} \boldsymbol{B}_{mc}$. These fuzzy values indicate the level to which each microphone also contains information of a different cluster and can be exploited for further separation tasks [14, 15]

6.4 Feature-Based Clustering

Alternative to the coherence-based method, the clusters can also be generated by comparing features extracted from the microphone signals. Ideally, these features extract the underlying dominant source and are not influenced by reverberation or interference. This paper handles the hand-crafted modulated Mel frequency cepstral coefficients (Mod-MFCC) based features proposed in [4, 16], and the speaker embeddings extracted from speaker verification (SpVer) deep neural networks, proposed in [8]. In both cases, the feature vectors are clustered using fuzzy C-means (FCM) clustering.

6.4.1 Mod-MFCC Features

First, the MFCCs are computed, and cepstral mean subtraction (CMS) is performed. CMS reduces the effect of reverberation and lets the features better focus on the underlying speech data [17, 18]. Taking the discrete Fourier transform (DFT) of the MFCCs, with a rectangular window of length L and modulation shift Q, generates the modulated MFCCs features. The modulation spectra are then averaged over time to account for the relatively big time shifts possible due to the inter-microphone distances in WASNs. The final features consist of two cepstral modulation ratios (CMRs) and the averaged modulation amplitude (AMA) [4, 16], each generated for 13 cepstral bins, resulting in a 39-dimensional feature vector.

6.4.2 Speaker Verification Features

Similar to [8], the paper takes the Emphasized Channel Attention, Propagation, and Aggregation Time Delay Neural Network (ECAPA-TDNN) [19] to generate speaker embeddings which, here, are used as clustering features. The network takes signals of arbitrary length and generates 192-dimensional speaker embeddings. ECAPA-TDNN extends the popular X-vector [20] system with an attentive statistics pooling layer, a speech-adapted version of Squeeze-Excitation (SE) [21] and multi-layer feature aggregation.

6.4.3 Fuzzy C-Means (FCM) Clustering

In contrast to the coherence-based method, both feature-based methods use the FCM algorithm [22] to cluster the microphones. Again $C = J + 1$ clusters will be generated. The algorithm makes use of cluster centers, $\mathcal{C}_c$, and fuzzy membership values (FMV), $\mu_{m,c}$. These FMV are interpreted similarly to those at the output of the NMF. The following loss is minimised in order to come to the final clusters :

$$\mathcal{L} = \sum_{c=0}^{C-1} \sum_{m=0}^{M-1} \mu_{m,c}^{\alpha} \delta(\mathcal{F}_m, \mathcal{C}_c)\,, \tag{6.7}$$

where $\mathcal{F}_m$ is the feature (either Mod-MFCC or SpVer), $\delta(\mathcal{F}_m, \mathcal{C}_c)$ is the distance between $\mathcal{F}_m$ and $\mathcal{C}_c$, and α is the fuzzy weighting exponent. In contrast to the previous work, we take the cosine distance: $\delta(\mathcal{F}_m, \mathcal{C}_c) = 1 - \frac{\mathcal{F}_m^T \mathcal{C}_c}{||\mathcal{F}_m||_2 ||\mathcal{C}_c||_2}$, with $\|.\|_2$ is the ℓ_2 norm of a vector. We found that the cosine distance outperforms the Euclidean for both feature types [23].

6.5 Evaluation and Results

To evaluate the performance of the different clustering methods, experiments similar to [6] were conducted: dry speech signals of 4s length are selected from the LibriSpeech clean speech database [24]. Subsequently, $M = 16$ microphones were placed in the SINS simulated living area [25] with $J = 2$ simultaneously active speakers. The SINS database consists of realistic reverberant room impulse responses (RIRs) of an apartment. The microphone placement is constrained so that at least 3 microphones lie within the critical distance of each source, as done in [4]. This paper also limits the scenarios to cases where one source is placed at random in the left half of the room, while the second source is placed in the right half, similar to [8]. Note that with this setup, the sources can still be relatively close to each other since we do not constrain the minimal distance, but in general, they will be relatively far apart from each other. Metrics are averaged over 200 different microphone and source positions. See an example setup in Figure 6.1. All signals are sampled at $f_s = 16$kHz, and the number of clusters is set to $M = J+1$, where the additional cluster represents a background (noise) cluster. Note that this is the same as done in the feature-based methods [4, 8], but different from the coherence based method, where $M = J$ was proposed to keep the comparison proper. Also, the number of sources is assumed known here, although this should normally also be estimated. However, this is out of the scope of this comparative paper.

In order to evaluate clustering performance, two classes of metrics are used as presented in [8]. The first handles the quality of the microphones in each cluster based on oracle knowledge of the underlying signals. The second looks at the

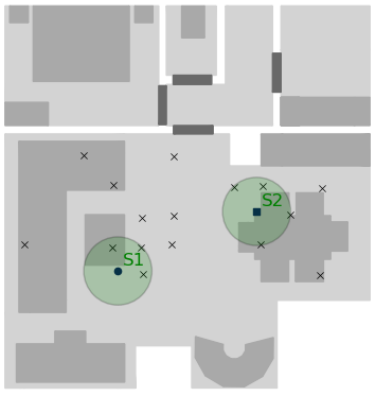


Figure 6.1: SINS apartment for a specific scenario. The solid dots indicate the location of the two sources, while the crosses are the microphone positions. The green circles indicate the critical distance region for each source ($d_{crit} = 0.68\,m$ for the room).

quality of the subsequent cluster-based source separation proposed in [14, 15]. The two classes are referred to as cluster metrics and separation metrics respectively.

6.5.1 LC3plus Codec

Besides the performance of the different clustering methods, it is important to consider the feasibility of the methods in bandwidth-limited WASNs. The feature-based methods only require a minimal bandwidth, since they only need to transmit their features to the central node. In contrast, the coherence method requires the entire signal from each node. To alleviate this requirement, the signals can be encoded prior to transmission to the central node. While this approach still imposes a heavier burden on the network, it's not without its benefits: many other applications require access to these signals, *e.g.*for beamforming.

This paper considers the extension of the low complexity communications codec (LC3), LC3plus, at its lowest bitrate, namely 16kbps (instead of 512kbps to send 32bit samples at 16kHz). LC3plus aims to transmit high-quality audio over wireless connections at reduced bandwidth/bitrates and is used in *e.g.*Bluetooth Low Energy (BLE) and Digital Enhanced Cordless Telecommunications (DECT) [1].

To evaluate the feasibility of the coherence-based methods, we let the individual sensor nodes (each with one microphone) encode their captured signal before sending it to the central access point. There the signals will be decoded and the coherence-based clustering will be carried out on these signals.

[1] https://www.iis.fraunhofer.de/en/ff/amm/communication/lc3.html

6.5.2 Clustering Metrics

The distribution of the direct-to-reverberant, interference, and noise ratio (DRINR) provides insight into whether the clustering favours microphones with a strong direct-path component and a good signal-to-interference and noise ratio. DRINR is defined as:

$$\text{DRINR}_{j,m} = \frac{\sum_n (x_{j,m}^{\text{dir}}(n))^2}{\sum_n (y_m(n) - x_{j,m}^{\text{dir}}(n))^2}\,, \tag{6.8}$$

and is calculated if microphone m is part of the source cluster j. Good clusters should have many microphones with high DRINRs while avoiding including low DRINR microphones. To assess this, DRINR histograms are plotted in Figure 6.2. The average number of microphones per cluster is also reported since this provides an indication of the spatial diversity within the cluster.

6.5.3 Separation Metrics

The main goal of clustering is to facilitate subsequent tasks such as source separation. Therefore, the quality of this separation indirectly reflects the clustering quality. Three metrics are used for this evaluation: the Source-to-Interference Ratio (SIR) [26], the Perceptual Evaluation of Speech Quality (PESQ) [27], and the Short-Time Objective Intelligibility (STOI) [28], where higher scores mean better performance.

Figure 6.3 plots these metrics for the different clustering methods (colors) and separation techniques (x-axis). The four techniques are: (1) initial mask-based separation (masks), (2) delay and sum beamforming (DSB), (3) fuzzy membership value aware DSB (FMVA_DSB) and (4) a postfilter applied on the DSB (postfilter). The dotted line represents the metric in case only the reference microphone for each source is picked, showing that the separation techniques indeed improve upon selecting the best microphone. The time-frequency (TF) mask is generated by comparing the amplitude of the STFT bins between all reference microphones of each cluster. The reference microphones are determined based on the highest FMVs within each cluster. A binary TF mask is then generated by selecting those STFT bins that have a higher amplitude than those for other clusters. Here a small temporal averaging is performed to account for the distances between microphones. These masks are applied to all microphones of the associated cluster, and a relative time delay is estimated, after which they are compensated for in the DSB. For the FMVA-DSB, the microphone signals contribute to the beamformed signal proportional to the FMV instead of being averaged. For the postfilter, the binary TF mask is computed *w.r.t.* the beamformed signals instead of the unprocessed microphone signals. For a detailed overview of these techniques, we refer to [14, 15]. Most important for this evaluation: a successful separation result requires good clusters,

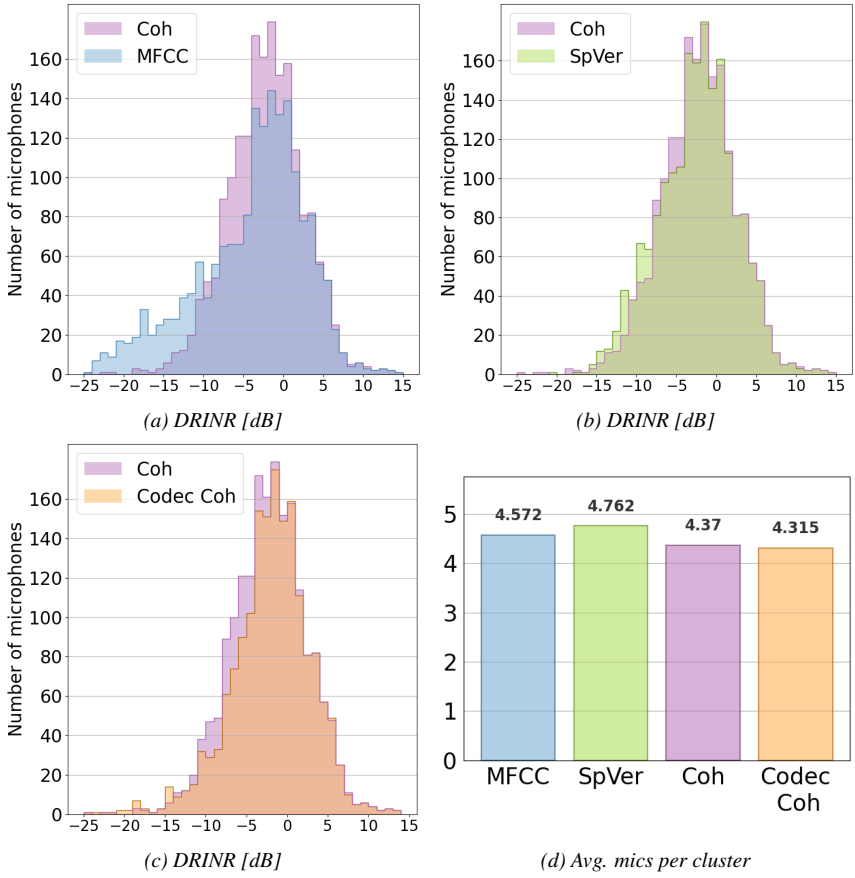


Figure 6.2: (a-c) Histograms of the direct-to-reverberant, interference, and noise ratio (DRINR). These are computed only for microphones that are part of a source cluster. (d) Average number of microphones per source cluster.

and can be degraded significantly with the inclusion of poor SNR microphones. Thus separation performance indirectly allows comparison of the cluster quality.

Note that for the separation, we assume that a central node has access to all the microphone signals. Even for the coherence-based method with the codec, the separation is done on the unencoded signals. This would in practice never happen, but is needed to keep the comparison fair. Speaker separation on encoded data is out of scope for this paper in order to focus on quantifying the quality of the clusters.

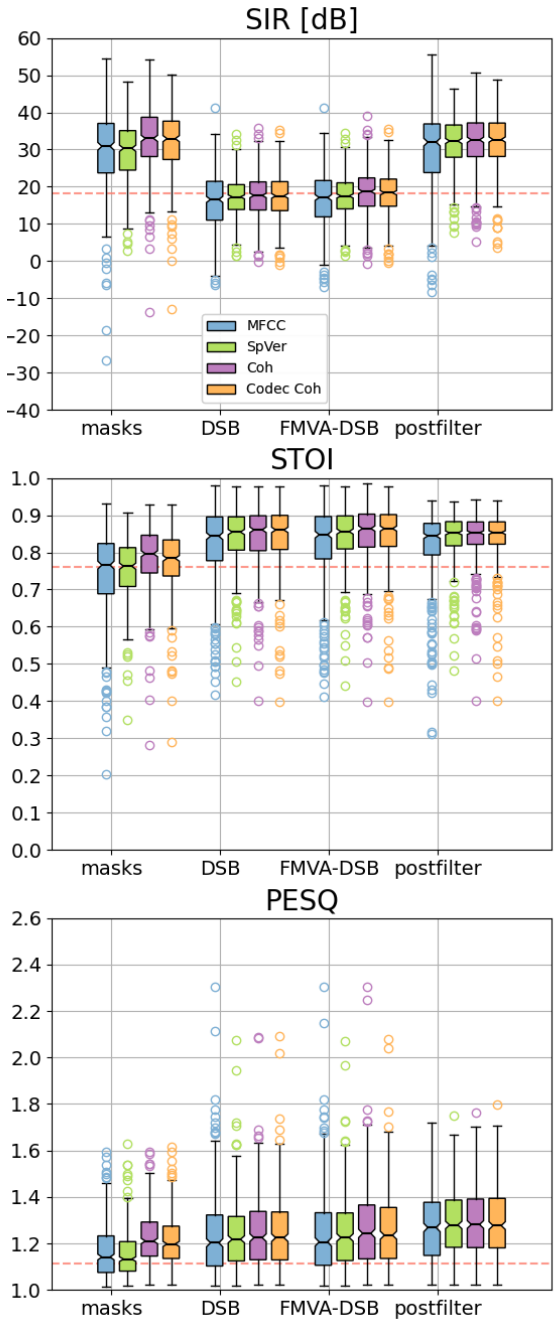


Figure 6.3: SIR[dB], PESQ and STOI for, Mod-MFCC and SpVer features, and coherence and coherence after LC3plus processing. The red dashed lines denote the mean of the optimal (oracle) microphones, showing optimal unprocessed performance.

6.6 Results and discussion

6.6.1 Coherence- v.s. Feature-Based Clustering

Figure 6.2b illustrates the distributions of the DRINRs for the coherence-based method and the SpVer features. The distribution is fairly similar, suggesting that the clustered microphones are either the same or of equivalent quality. At very high DRINRs, the distribution is even identical. However, there are still slight variations in their distributions: around -20 dB DRINR, the coherence-based method picks up slightly more outliers. In contrast, the SpVer method includes more microphones within the range of -15dB and -10dB DRINR, which are possibly not the most useful microphones, depending on how close the interferer is. From -8dB to 0dB DRINR, the coherence-based method includes more microphones with moderately good DRINRs. Additionally, Figure 6.2d shows that the speaker embedding method includes more microphones on average. This, combined with the skew towards lower DRINRs suggests that the clusters from the coherence-based method should be preferred by a slight margin for these scenarios.

If we look at the initial mask-based speaker separation metrics in Figure 6.3, we notice that the coherence-based clusters clearly outperform the feature-based methods. This can be explained by the superior reference microphone selection by the coherence and NMF combination compared to the feature and FCM combination. Indeed, FCM tries to find the cluster centre that best resembles the average signal of that cluster, giving the highest FMV to the microphone closest to that cluster centre. In contrast, the coherence of all microphones of the same cluster will be highest on average towards the microphone closest to the source, resulting in the highest FMV after NMF.

However, for the subsequent separation steps, the choice of reference microphone plays no role in the separation quality, as long as the SIR of the masked signals is high enough to correctly estimate the relative delays between microphones. The gap between the feature- and coherence-based methods lessens. Nevertheless, the coherence-based method still slightly outperforms the SpVer features and clearly outperforms the Mod-MFCC features.

Furthermore, the distribution in Figure 6.2a indicates that the coherence-based method clearly outperforms the method based on MFCCs, since it picks up significantly more microphones with high DRINRs and fewer with lower DRINRs. This is in line with the findings in [8], where a shoe-box simulated room was used.

6.6.2 Effect of LC3plus Lossy Encoding

The objective here was to determine the extent to which coherence-based cluster quality deteriorates under the non-linear, lossy encoding of the LC3plus codec at 16kbps. The results of the DRINRs (see Figure 6.2c) show a remarkable resilience

of the coherence-based method when confronted with LC3plus encoded signals. The encoded signals do deliver some extra outliers below -15dB DRINR, and include fewer microphones from the range -10dB to 0dB DRINR. Although the former is not a good property, the extent to which this happens is rather limited. The latter however is hard to judge and it is unclear which clustering is superior. On one hand, including more microphones increases the spatial diversity, on the other hand, it also reduces the average SNR of the clustered microphones. Figure 6.2d also confirms that the unencoded signals deliver slightly larger clusters on average.

This is somewhat unexpected since coherence looks for linear dependencies between the microphone signals, and LC3plus encodes the individual signals in a lossy and non-linear way. This suggests that LC3plus encodes the signals in similar ways for each microphone signal, thus keeping same the linear dependency between the signals present before encoding.

The separation metrics in Figure 6.3 illustrate, similarly to the DRINR distribution, minimal performance degradation. All metrics show that the performance degrades minimally. Only the initial masks-based separation degrades more than the other separation methods, indicating that the encoded clustering cannot recognise the optimal reference microphone as well as the unencoded clustering. However, this still exceeds the performance of the fuzzy C-means clustered signal-specific features.

Although this paper only reports the results of the lowest bitrate available in LC3plus, experiments with higher bitrates also do not degrade the clusters significantly and thus keep the linear dependencies between signals. This is not surprising, since higher bitrates require less lossy encoding.

This result establishes the coherence-base method as a viable option in bandwidth-limited WASNs with the help of a codec, requiring minimal computational power at each node and making all the individual microphone signals available for subsequent processing. Note that the optimal solution is still very application-specific, considering factors such as bandwidth and processing power requirements. For example, the lower bandwidth requirements of the feature-based clusters can be exploited by first identifying which microphone signals are needed for further processing, before sending those to the central node. *e.g.*the speaker verification only needs 192 features, which with a 32-bit representation results in ˜6kb that needs to be sent to the central node per microphone. In contrast, a 4-second segment at 16kpbs still requires 64kb.

6.7 Conclusions

Unveiled by the cluster metrics and speaker separation metrics, the coherence-based clustering method has demonstrated a slight edge over the speaker verification method. Similarly, the evaluation confirmed that Mod-MFCC features perform worse than SpVer features and thus also the coherence-based method.

A possible downside to the coherence-based clustering method for bandwidth-limited WASNs is that all audio signals need to be transmitted to a central node, while the feature-based methods only need to share their feature vectors with a central node in order to perform the clustering. Therefore, the coherence-based method was evaluated on signals that were encoded and decoded with the reduced-bitrate codec LC3plus.

The evaluation metrics of the coherence-based clustering method with LC3plus processing demonstrate that the method remains robust in the presence of this lossy encoding. This suggests that, even though the signals are individually encoded in a non-linear manner, the linear relations between different signals are still preserved. Therefore, LC3plus encoding only slightly changes the quality of clustering. If in addition, the subsequent processing task needs access to the microphone signals, the signal transmission is far from wasted, making the encoded coherence-based method a viable option in these bandwidth-limited systems. Nevertheless, the feature-based methods could make a first selection of which microphone signals are important to be fully sent to the central node, avoiding redundant or unnecessary signals, and thereby further optimising bandwidth usage.

Further, we found that the current method to select the reference microphone from the output of the FCM algorithm, as reported in previous work, is imperfect and should be optimised in future work. Additionally, future research could evaluate the quality of source separation with LC3plus-processed signals. This analysis would provide valuable insights into the minimal bitrate, and thus bandwidth requirements in WASNs for sufficient audio quality. Lastly, more clustering techniques need to be included in the comparative study.

Acknowledgment

This work is supported by the Research Foundation - Flanders (FWO) under grant number G081420N and imec.ICON: BLE2AV (support from VLAIO). Partners: Imec, Televic, Cochlear, and Qorvo.

References

[1] A. Bertrand. *Applications and trends in wireless acoustic sensor networks: A signal processing perspective*. In 2011 18th IEEE symposium on communications and vehicular technology in the Benelux (SCVT), pages 1–6. IEEE, 2011.

[2] I. Himawan, I. McCowan, and S. Sridharan. *Clustered blind beamforming from ad-hoc microphone arrays*. IEEE Transactions on Audio, Speech, and Language Processing, 19(4):661–676, 2010.

[3] S. Pasha, Y. X. Zou, and C. Ritz. *Forming ad-hoc microphone arrays through clustering of acoustic room impulse responses*. In 2015 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP), pages 84–88. IEEE, 2015.

[4] S. Gergen, A. Nagathil, and R. Martin. *Classification of reverberant audio signals using clustered ad hoc distributed microphones*. Signal Processing, 107:21–32, 2015.

[5] A. Nelus, R. Glitza, and R. Martin. *Estimation of microphone clusters in acoustic sensor networks using unsupervised federated learning*. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 761–765. IEEE, 2021.

[6] A. Nelus, R. Glitza, and R. Martin. *Unsupervised Clustered Federated Learning in Complex Multi-source Acoustic Environments*. In 2021 29th European Signal Processing Conference (EUSIPCO), pages 1115–1119. IEEE, 2021.

[7] L. Becker, A. Nelus, R. Glitza, and R. Martin. *Accelerated Unsupervised Clustering in Acoustic Sensor Networks Using Federated Learning and a Variational Autoencoder*. In 2022 International Workshop on Acoustic Signal Enhancement (IWAENC), pages 1–5. IEEE, 2022.

[8] S. Kindt, J. Thienpondt, and N. Madhu. *Exploiting Speaker Embeddings for Improved Microphone Clustering and Speech Separation in ad-hoc Microphone Arrays*. In ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1–5. IEEE, 2023.

[9] A. J. Muñoz-Montoro, P. Vera-Candeas, and M. G. Christensen. *A Coherence-based Clustering Method for Multichannel Speech Enhancement in Wireless Acoustic Sensor Networks*. In 2021 29th European Signal Processing Conference (EUSIPCO), pages 1130–1134. IEEE, 2021.

[10] C. Ding, X. He, and H. D. Simon. *On the equivalence of nonnegative matrix factorization and spectral clustering*. In Proceedings of the 2005 SIAM international conference on data mining, pages 606–610. SIAM, 2005.

[11] M. Schnell, E. Ravelli, J. Büthe, M. Schlegel, A. Tomasek, A. Tschekalinskij, J. Svedberg, and M. Sehlstedt. *LC3 and LC3plus: The new audio transmission standards for wireless communication*. In Audio Engineering Society Convention 150. Audio Engineering Society, 2021.

[12] P. Welch. *The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms*. IEEE Transactions on audio and electroacoustics, 15(2):70–73, 1967.

[13] D. Lee and H. S. Seung. *Algorithms for Non-negative Matrix Factorization*. In T. Leen, T. Dietterich, and V. Tresp, editors, Advances in Neural Information Processing Systems, volume 13. MIT Press, 2000. Available from: https://proceedings.neurips.cc/paper/2000/file/f9d1152547c0bde01830b7e8bd60024c-Paper.pdf.

[14] S. Gergen, R. Martin, and N. Madhu. *Source separation by feature-based clustering of microphones in ad hoc arrays*. In 2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC), pages 530–534. IEEE, 2018.

[15] S. Gergen, R. Martin, and N. Madhu. *Source separation by fuzzy-membership value aware beamforming and masking in ad hoc arrays*. In Speech Communication; 13th ITG-Symposium, pages 1–5. VDE, 2018.

[16] S. Gergen and R. Martin. *Estimating source dominated microphone clusters in ad-hoc microphone arrays by fuzzy clustering in the feature space*. In Speech Communication; 12. ITG Symposium, pages 1–5. VDE, 2016.

[17] F. Bimbot, J. F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-García, D. Petrovska-Delacrétaz, and D. A. Reynolds. *A tutorial on text-independent speaker verification*. EURASIP Journal on Advances in Signal Processing, 2004(4):1–22, 2004.

[18] P. N. Garner. *Cepstral normalisation and the signal to noise ratio spectrum in automatic speech recognition*. Speech Communication, 53(8):991–1001, 2011.

[19] B. Desplanques, J. Thienpondt, and K. Demuynck. *ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN based speaker verification*. In Interspeech2020, pages 3830–3834. International Speech Communication Association (ISCA), 2020.

[20] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur. *X-Vectors: Robust DNN Embeddings for Speaker Recognition*. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5329–5333, 2018. doi:10.1109/ICASSP.2018.8461375.

[21] J. Hu, L. Shen, and G. Sun. *Squeeze-and-Excitation Networks*. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7132–7141, 2018. doi:10.1109/CVPR.2018.00745.

[22] J. C. Bezdek, R. Ehrlich, and W. Full. *FCM: The fuzzy c-means clustering algorithm*. Computers & geosciences, 10(2-3):191–203, 1984.

[23] S. Kindt, J. Thienpondt, L. Becker, and N. Madhu. *Robustness of ad hoc microphone clustering using speaker embeddings: evaluation under realistic and challenging scenarios*. EURASIP Journal on Audio, Speech, and Music Processing, 2023(1):46, 2023.

[24] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. *Librispeech: an asr corpus based on public domain audio books*. In 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP), pages 5206–5210. IEEE, 2015.

[25] R. Glitza, L. Becker, A. Nelus, and R. Martin. *Database of Simulated Room Impulse Responses for Acoustic Sensor Networks Deployed in Complex Multi-Source Acoustic Environments.* In EUSIPCO, 2023.

[26] E. Vincent, R. Gribonval, and C. Févotte. *Performance measurement in blind audio source separation*. IEEE transactions on audio, speech, and language processing, 14(4):1462–1469, 2006.

[27] 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., volume 2, pages 749–752, 2001.

[28] 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, pages 4214–4217, 2010.

7

Enhanced Deep Speech Separation in Clustered Ad Hoc Distributed Microphone Environments

In this chapter, the clustering from Chapters 5 and 6 will be exploited as relative location information for deep speech separation. Chapter 3 also dealt with location informed separation, but the type of location information is drastically different. In Chapter 3, the DOAs of the speakers were inputted in the form of a multi-hot vector or expected phase difference features. For this chapter, the location information indicates which microphones are closest to each of the speakers. The method of introducing this information is therefore also drastically different. Instead of providing additional input features, the method selects which microphones are processed together. This selection will prove to be invaluable.

⋆ ⋆ ⋆

Jihyun Kim*, Stijn Kindt*, Nilesh Madhu, Hong-Goo Kang

Accpeted in the 25th Interspeech Conference (Interspeech 2024).

Abstract Ad-hoc distributed microphone environments, where microphone locations and numbers are unpredictable, present a challenge to traditional deep learning models, which typically require fixed architectures. To tailor deep learning

* Equal contributions, shared first author

models to accommodate arbitrary array configurations, the Transform-Average-Concatenate (TAC) layer was previously introduced. In this work, we integrate TAC layers with dual-path transformers for speech separation from two simultaneous talkers in realistic settings. However, the distributed nature makes it hard to fuse information across microphones efficiently. Therefore, we explore the efficacy of blindly clustering microphones around sources of interest prior to enhancement. Experimental results show that this deep cluster-informed approach significantly improves the system's capacity to cope with the inherent variability observed in ad-hoc distributed microphone environments.

7.1 Introduction

In acoustic sensor networks (ASNs), multiple microphones can be arbitrarily distributed throughout a given space. This distributed configuration provides extensive spatial coverage in contrast to compact microphone arrays [1]. ASNs are also becoming more common in daily life, with the growing number of devices equipped with one or multiple microphones, like smartwatches, smartphones, laptops and smart glasses. While speech processing [2–4], speaker localisation [5, 6] and speaker verification [7] have made advancements utilising this extra spatial information, there remains much to be explored in this field to *fully* capitalise on the possibilities it opens up.

ASNs, particularly those deployed in an ad-hoc manner, have extra challenges associated with them. Firstly, the number of microphones and their respective positions are not known and may vary throughout its operation due to environmental changes. These changes can include devices entering or leaving the environment, or moving within the space. Secondly, due to the potentially widely-distributed nature of the microphones, the same speech signal can be captured at two different microphones at very different time instances. Also, the microphones operated on independent clocks, leading to discrepancies in sample rate offsets (SROs) and sample time offsets (STOs). Lastly, all the microphones could have very different characteristics, i.e. frequency response and directivity. While the latter two can be robustly solved by other methods [8, 9], the core challenge of speaker separation endures.

One solution previously proposed by Gergen *et al.*[10, 11] is to first cluster the microphones either around the speakers or into a background (noise) cluster. Subsequently, cluster information is leveraged within classical signal enhancement frameworks, demonstrating superiority over methods like optimal microphone selection. The usefulness of clustering has also been shown in [12], where incorporating microphones that are far away from a target speaker (from outside the cluster) can degrade the result. To date, cluster-based separation techniques have only been investigated within the realm of classical signal processing. However, we hypoth-

esise that deep neural network-based separation methods could also benefit from the clustering, potentially surpassing the performance of classical methods.

The Transform-Average-Concatenate (TAC) layer[13] was previously proposed for deep, array-agnostic, *compact* array processing. This layer is introduced between blocks that individually process the inputs for each microphone channel and, consequently, acts as the information sharing layer between microphone channels in a permutation and number invariant manner. Thus, it can handle the challenge of unknown array geometries. TAC has been successfully combined different architectures, e.g., dual-path recurrent neural network (DPRNN) [14] and VarArray [15], where Conformers [16] are used for time-frequency processing. Alternative array-agnostic methods, like [17], use multi-head cross-attention to share information across microphones. However, only limited research [17, 18] has been conducted specifically on distributed microphone setups. Their investigation revealed both the potential benefits and the intricate challenges associated with employing variably located microphones for speech processing. A significant gap identified is the need for better methods to utilise the spatial diversity of the microphones in a more informed manner.

This paper proposes a novel approach that incorporates the blind microphone clustering techniques into *cluster-informed*, array-agnostic deep learning methodologies. In short:

1. Initially, in the ad-hoc distributed microphone environment, a blind spatial-statistics-based clustering approach [19] is employed to cluster microphones around the active speakers. Additionally, the clustering also estimates a *pseudo* reference microphone, where the target speech should be the most dominant in all microphones of that cluster.

2. Then a deep learning-based network exploits spatial information from all microphones within each speaker-dominated cluster to extract the underlying target speech. As we demonstrate, the optimal configuration exploits, in addition to spatial information from all microphones within the cluster, the benefit offered by selecting a robust reference microphone.

Additionally, we propose a training data generation method to simulate clustered data without actually executing the clustering. This is important to save considerable training time and ensures that the training is independent of the specific clustering algorithm employed. The deep separation method will be compared to the classical processing methods and ablation studies will show the effectiveness of the proposed method compared to alternative deep learning structures that do not use the cluster information to its fullest potential. The paper will first overview the classical techniques, where a brief explanation of the clustering and separation method is given in Section 7.2. Then the proposed deep architecture is explained in Section 7.3, and evaluated in Section 7.4. Section 7.5 concludes the paper.

7.2 Classical Methods

7.2.1 Clustering

Ensuring robust clustering is essential for distributed microphone techniques [20, 21]. Based on the findings of the comparative study presented in [22], the full bandwidth, coherence-based clustering method [19] is chosen. This algorithm, represented by the first two steps in Figure 7.1, uses the pairwise magnitude squared coherence between all M microphones as features, denoted by $\mathcal{F}$, and organises them into a matrix $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{M \times M}$. Then, non-negative matrix factorisation (NMF) [23] is utilised to cluster the microphones by decomposing this matrix as: $\boldsymbol{\mathcal{C}} = \boldsymbol{B}\boldsymbol{B}^T \odot (\mathbf{1} - \boldsymbol{I}) + \boldsymbol{I}$, where $\odot$ is the element-wise (Hadamard) product, $\boldsymbol{I}$ denotes the identity matrix, $\mathbf{1}$ is the all-ones matrix and $\boldsymbol{B} \in \mathbb{R}^{M \times C}$ is the cluster matrix. This matrix contains all fuzzy membership values (FMVs) of each microphone towards each cluster, where $\boldsymbol{B}_{mc}$ represents the contribution of microphone m to cluster c. For hard clustering, microphones are then attributed to the cluster where their contribution is highest. Additionally, for each cluster, a reference microphone is identified as the microphone with the highest fuzzy membership value for that cluster. The number of clusters C is one greater than the number of speakers, where the last cluster collects microphones mostly dominated by noise and reverberations. Both the (hard) cluster and the reference microphone will prove invaluable for *informed* speech separation.

7.2.2 Separation

In Figure 7.1, the relation between the different classical cluster based separation methods are shown: (1) initial masks are estimated by comparing the amplitude of the short-time Fourier transform (STFT) bins across all reference microphones within each cluster, exploiting sparsity and disjointness of speech [24]. (2) Then, relative time delays $\widehat{\tau}_{m,c}$ are estimated on the masked cluster signals, and used for delay and sum beamforming (DSB) of all clustered microphone signals. (3) A microphone weighting based on the fuzzy values can be included in the fuzzy membership value aware DSB (FMVA-DSB). (4) Lastly, a postfilter can obtained by comparing the beamformed signal of each cluster. For a detailed overview, we refer to [10, 11]. All the methods succeed in a better foregrounding of target speakers, but the resulting audio quality is poor. The masks of initial and postfilter are binary, distorting the signal, and the simple beamformers cannot sufficiently cancel the interferer.

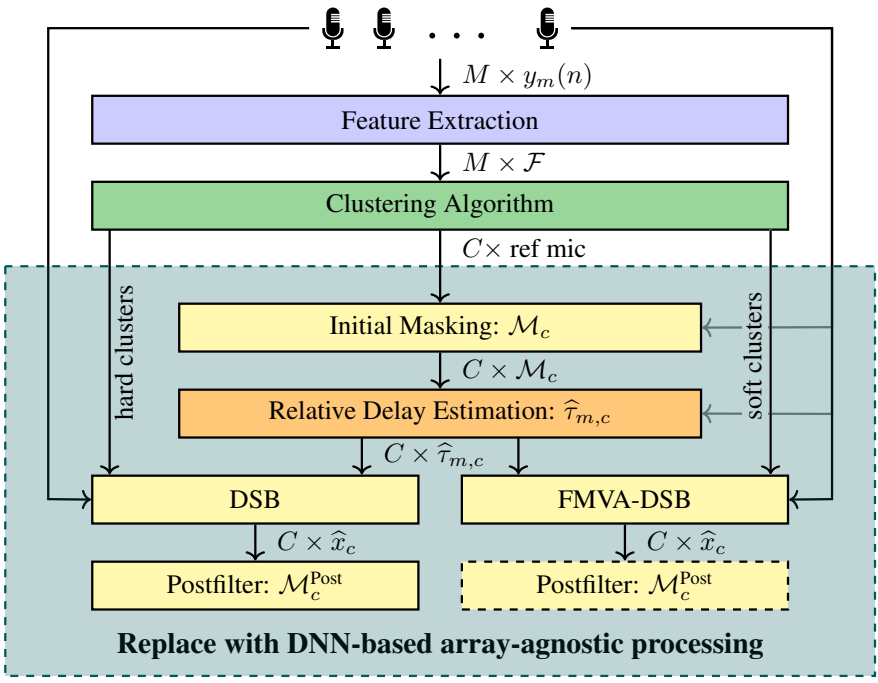


Figure 7.1: Scheme of the classical cluster-based source separation method. The grayed region is replaced by our proposed separation.

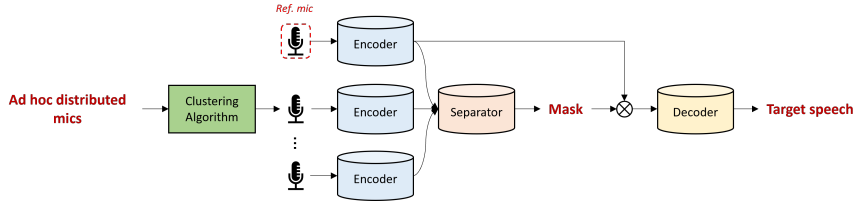


Figure 7.2: Overall system architecture of the proposed deep, array-agnostic, target extraction approach. The reference microphone is identified from the clustering. For separation, the mask is applied to the embedding of the reference microphone.

7.3 Proposed Method

As illustrated in Figure 7.2, our proposed method clusters ad-hoc distributed microphones around speakers and selects a reference microphone for each cluster. This is done as described in Section 7.2.1. Microphones of each cluster are then processed through a deep learning-based separation network, which consists of Encoder, Separator and Decoder.

By clustering first, each cluster can be independently processed - removing the

need for separation techniques such as permutation invariant training (PIT) [25]. We only need to extract the *dominant* source of each cluster with a multi-channel time-domain network. The network architecture is based on the VarArray structure [15], where Conformers are swapped with dual-path transformer networks (DPTNets) [26]. The use of the DPTNet allows for a computationally efficient method to process both local and global information, leading to a comprehensive and accurate representation of the acoustic scene.

Encoder. The Encoder transforms raw multi-channel speech signals $x \in \mathbb{R}^{M\times 1\times T}$ into a high-dimensional feature $h \in \mathbb{R}^{M\times N\times T}$ using a 1-D convolution layer, where T is the number of time frames N is the number of convolutional kernels of the encoder (and thus the number of features) and M the number of microphones inputted to the encoder. Additionally, the Encoder incorporates a segmentation strategy derived from DPTNet, allowing for more precise handling of both local and global dependencies within the speech signal. It splits the hidden feature h into overlapped chunks of length K with a hop size of $K/2$. The hidden feature h is thus a 4-D tensor $h_0 \in \mathbb{R}^{M\times N\times K\times P}$, where P is the number of chunks. This design choice ensures that the network both preserves the detailed temporal structure of the speech signal and increases its receptive field. This is essential to compensate for the relatively long time delays in widely distributed microphone settings.

Separator. The separator combines TAC layers and DPTNets within its processing chain for spatial and temporal processing respectively. Similar to the structure of VarArray, 3 DPTNets are interleaved with 2 TAC layers, followed by a mean pooling and 2 DPTNets. The first DPTNets process each microphone individually, where the TAC layers combine the microphone information. Mean pooling reduces computational complexity for the following (single channel) DPTNets. The separator produces a mask, which is applied to an encoder embedding. Here the suitability of the clustering, more specifically the indication of a reference microphone, is again clear. If this information is present, the hidden feature of the reference microphone can be selected for masking.

We propose further changing the mean pooling layer by selecting the embedding from the reference microphone. Firstly, this embedding should be the most dominated by the target speech. Secondly, it reduces the computational cost since no DPTNets operations are applied on the non reference embeddings. The proposed method is depicted in Figure 7.3. We will term this proposed style, while the previously described networks VarArray style in the ablation studies of Section 7.4.3.

Decoder. The Decoder reconstructs the separated speech signals from the enhanced high-dimensional latent representation. In a process that mirrors the encoder, the decoder employs overlap and add, followed by a 1-D transposed convolution layer, to transform the enhanced hidden feature back into the time-domain

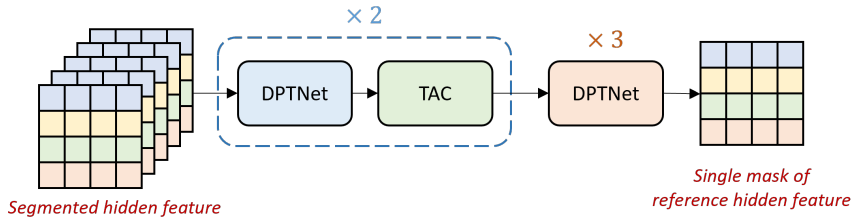


Figure 7.3: Detail structure of the separator. For multi-microphone processing, a single module operates in parallel across the microphones.

yielding clear, distinct speech tracks.

7.4 Experimental Evaluation

In this section, we want to show that deep networks improve upon classical separation methods and that cluster information is essential for good separation. Therefore, next to comparison with the classical methods, we include ablation studies where all microphones are used as input to the neural network (*unclustered version*), an ablation study where only the reference microphone (*single microphone*) is used as input, and study the difference between VarArray style and the proposed style.

The unclustered version is trained with Permutation Invariant Training (PIT) loss to separate the different speakers. This is needed because there is no method to determine which speaker should correspond to which output channel. Since no reference mic is known, a random microphone is selected of which the encoding embeddings are taken for the decoder. Initial experiments showed that averaging over the embeddings performs worse. This highlights why clustering in widely distributed microphones is beneficial.

The ablation study with the single microphone method, where no TAC layers are needed, indicates whether the spatial diversity, provided by the multiple microphones within a cluster, is valuable. The VarArray style ablation study reveals whether incorporating the reference microphone information within the network structure improves the final result.

We will begin by outlining the datasets used for training and evaluation, as well as the specific parameter choices, before delving into the results.

7.4.1 Dataset

To train our network, we utilised the WSJ0-2mix [27] clean speech dataset, and convolved them with the shoebox room impulse responses (RIRs), generated by the image source model of gpuRIR [28]. White noise was added at SNRs uniformly

sampled between 0 dB and 20 dB. A variety of different rooms (dimensions and reverberation times) are generated to promote generalisability, totalling 10,080 different scenarios. For this work, the simulation was limited to cases where the two speakers were located in different halves of the room. A total of 16 microphones were simulated for each scenario, where for each source there are at least 3 microphones within its critical distance, consistent with previous cluster based separation work [21]. We iterate over all RIRs for each epoch and select a random clean speech sample on the fly to increase diversity. Also, for each batch, a random selection of microphones – between 8 and 16 microphones – is chosen, to expose the network to different numbers of microphones and increase the total number of possible scenarios. However, it is ensured that the microphones within the critical distance are kept during selection.

Clustered Training Dataset. However, clustering on the fly would waste valuable training time since NMF is an iterative method. Also, this might make the network dependent on the clustering algorithm. To alleviate these problems, a second dataset of clustered RIRs is generated, where microphone positions are simulated as if they could have originated from clustering. Two sources are still sampled in different room halves, but the microphones are no longer sampled in the whole room. Three microphones are still placed within the critical distance of the speaker, and one is selected as the reference microphone. 4 other microphones are simulated in a $2m \times 2m$ square centred around the speaker. During training, a random number of microphones between 3 and 7 is chosen for generalisability to unknown microphone numbers. The unclustered dataset is utilised to train the unclustered version, while the clustered dataset is used to train all other networks.

Evaluation Dataset. To assess the real-world applicability of the model, the realistic SINS dataset [29], simulated with a CATT model, is used. The evaluation set is done similarly to [21, 22]. Two speaker positions are selected in opposite halves of the room and 16 microphones are distributed over the room. For each source, at least 3 microphones are within the critical distance. If the speakers are both sampled towards the middle of the room, they *could* still be very closely spaced. Dry speech is taken from the LibriSpeech dataset [30]. White noise is added, at an SNR of 10 dB with respect to the middle of the room. The actual SNR at individual microphones can differ greatly.

Good performance on real clustered data would validate the clustered training dataset. Also, since this dataset differs significantly in realism and sound sources, it can demonstrate the model's performance and generalisation capability for environments that closely mimic actual speech separation challenges.

7.4.2 Experiment Setup

For the Encoder and Decoder, we selected a kernel size of 8 samples with a stride of 50%. We use a segment size K of 250 on the segmentation for dual-path processing. We set the feature dimension of the separation network to be $N = 64$. We use 4 attention heads on each transformer layer in the DPTNet. The training criterion we used is the Scale Invariant Signal to Distortion Ratio (SI-SDR) [31] loss We used Adam optimiser and the training process began with an initial learning rate of 0.125, with a strategy to halve the learning rate if the validation loss doesn't decrease for three epochs. The total number of parameters in our proposed model is 2.23 M.

7.4.3 Experiment Results

To assess the performance, we employed three objective metrics: Scale Invariant Signal to Distortion Ratio (SI-SDR) [31], Perceptual Evaluation of Speech Quality (PESQ) [32], and Short-Time Objective Intelligibility (STOI) [33]

Figure 7.4 shows the results of the different separation methods. The *reference point* is given by the metrics computed on the *unprocessed* reference microphone signals. Firstly, it is clear, from the SI-SDR and PESQ, that combining information across *all* microphones (unclustered version) does not perform well. This indicates that the general, uninformed nature of TAC – which combines features from all microphone signals similarly, independent of the underlying speaker or background noise dominance at that microphone – does not suit distributed scenarios. Additionally, the lack of a good selection mechanism for hidden features on which the mask is applied, makes it hard for the network to generalise to other situations: the SI-SDR performance is very poor even though it was trained to maximise this metric.

Picking the unprocessed microphone closest to each speaker – the reference microphone in the clustering algorithm – outperforms the unclustered deep learning method and shows decent intelligibility (STOI). However, the output can be further improved. The classical methods do indeed increase the performance on all metrics, except for the initial masking – which is anyway mainly used for a robust, relative time delay estimation.

Using cluster informed deep learning algorithms significantly outperforms the classical methods. Using the reference microphone as input for a single channel model, where only DPTNets are sequentially applied, gives a big performance boost, most notably the big increase in PESQ. However, the method does not exploit the spatial diversity provided by the clustered microphones. When considering the inputs from clustered microphones, the metrics show that, unlike the unclustered version, TAC is effective. All microphones are dominated by the same source, removing target ambiguity. The performance of the clustered methods also

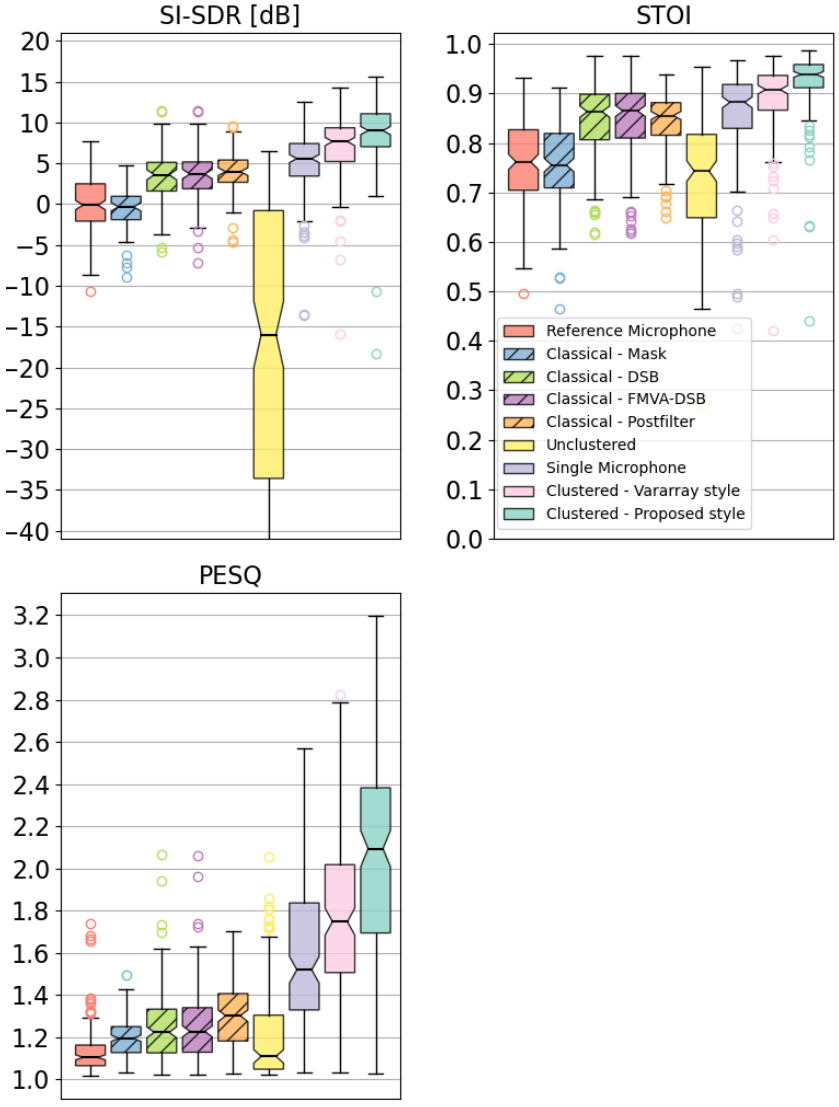


Figure 7.4: SI-SDR, STOI and PESQ for the different separation methods (higher is better)

supports the validity of the proposed data generation scheme.

The results also show that it is worthwhile to let the network prioritise the embeddings from the reference microphone. VarArray style averages the features over all the microphones before continuing with the single channel portion of the network. The proposed style takes the reference microphone as input for the single-channel portion of the network. This simple information inclusion in the design increases the performance significantly on all three metrics.

Specific scenarios, where the clusters are plotted and audio of the different separation techniques are present, can be found at https://aspire.ugent.be/demos/INTERSPEECH2024SK/.

7.5 Conclusion

In this paper, we introduced a novel approach for speech separation in ad hoc distributed microphone environments, combining coherence-based clustering methods with deep learning networks. Our experiments on realistically simulated RIRs show that it is essential to include cluster information in deep learning separation networks. More so, also including the reference microphone – a byproduct of the clustering method – further enhances the method. Conversely, the deep learning based separation gives a significant boost to the separation compared to classical methods. This highlights the benefits of combining traditional signal processing techniques with modern deep learning for speech processing tasks in real-world scenarios. Additionally, an efficient data generation paradigm to simulate clustered data was proposed for training such frameworks.

Future work could further increase the information the networks get from the clustering, by incorporating cross cluster information within the design of the network.

Acknowledgment

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

References

[1] A. Bertrand. *Applications and trends in wireless acoustic sensor networks: A signal processing perspective*. In 2011 18th IEEE symposium on communications and vehicular technology in the Benelux (SCVT), pages 1–6. IEEE, 2011.

[2] M. Souden, K. Kinoshita, M. Delcroix, and T. Nakatani. *Location feature integration for clustering-based speech separation in distributed microphone arrays*. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 22(2):354–367, 2013.

[3] D. Kim, S.-W. Chung, H. Han, Y. Ji, and H.-G. Kang. *HD-DEMUCS: General Speech Restoration with Heterogeneous Decoders*. In Proc. INTERSPEECH 2023, pages 3829–3833, 2023. doi:10.21437/Interspeech.2023-1642.

[4] J. Kim and H.-G. Kang. *Contrastive Learning based Deep Latent Masking for Music Source Separation*. In Proc. INTERSPEECH 2023, pages 3709–3713, 2023. doi:10.21437/Interspeech.2023-1723.

[5] S. Kindt, A. Bohlender, and N. Madhu. *2d acoustic source localisation using decentralised deep neural networks on distributed microphone arrays*. In Speech Communication; 14th ITG Conference, pages 1–5. VDE, 2021.

[6] H. Han and N. Kumar. *A cross-talk robust multichannel VAD model for multiparty agent interactions trained using synthetic re-recordings*. In 2024 Hands-free Speech Communications and Microphone Arrays (HSCMA), 2024.

[7] D. Cai and M. Li. *Embedding aggregation for far-field speaker verification with distributed microphone arrays*. In 2021 IEEE spoken language technology workshop (SLT), pages 308–315. IEEE, 2021.

[8] T. Gburrek, J. Schmalenstroeer, and R. Haeb-Umbach. *On Synchronization of Wireless Acoustic Sensor Networks in the Presence of Time-Varying Sampling Rate Offsets and Speaker Changes*. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 916–920, 2022. doi:10.1109/ICASSP43922.2022.9746284.

[9] A. Chinaev, N. Knaepper, and G. Enzner. *Long-Term Synchronization of Wireless Acoustic Sensor Networks with Nonpersistent Acoustic Activity Using Coherence State*. In ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1–5. IEEE, 2023.

[10] S. Gergen, R. Martin, and N. Madhu. *Source separation by feature-based clustering of microphones in ad hoc arrays*. In 2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC), pages 530–534. IEEE, 2018.

[11] S. Gergen, R. Martin, and N. Madhu. *Source separation by fuzzy-membership value aware beamforming and masking in ad hoc arrays*. In Speech Communication; 13th ITG-Symposium, pages 1–5. VDE, 2018.

[12] I. Himawan, I. McCowan, and S. Sridharan. *Clustered blind beamforming from ad-hoc microphone arrays*. IEEE Transactions on Audio, Speech, and Language Processing, 19(4):661–676, 2010.

[13] Y. Luo, Z. Chen, N. Mesgarani, and T. Yoshioka. *End-to-end microphone permutation and number invariant multi-channel speech separation*. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6394–6398. IEEE, 2020.

[14] Y. Luo, Z. Chen, and T. Yoshioka. *Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation*. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 46–50. IEEE, 2020.

[15] T. Yoshioka, X. Wang, D. Wang, M. Tang, Z. Zhu, Z. Chen, and N. Kanda. *VarArray: Array-geometry-agnostic continuous speech separation*. In ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6027–6031. IEEE, 2022.

[16] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, et al. *Conformer: Convolution-augmented Transformer for Speech Recognition*. Interspeech 2020, 2020.

[17] D. Wang, Z. Chen, and T. Yoshioka. *Neural Speech Separation Using Spatially Distributed Microphones*. INTERSPEECH 2020, pages 2467–2471, 2020.

[18] D. Wang, T. Yoshioka, Z. Chen, X. Wang, T. Zhou, and Z. Meng. *Continuous speech separation with ad hoc microphone arrays*. In 2021 29th European Signal Processing Conference (EUSIPCO), pages 1100–1104. IEEE, 2021.

[19] A. J. Muñoz-Montoro, P. Vera-Candeas, and M. G. Christensen. *A Coherence-based Clustering Method for Multichannel Speech Enhancement in Wireless Acoustic Sensor Networks*. In 2021 29th European Signal Processing Conference (EUSIPCO), pages 1130–1134. IEEE, 2021.

[20] S. Gergen, A. Nagathil, and R. Martin. *Classification of reverberant audio signals using clustered ad hoc distributed microphones*. Signal Processing, 107:21–32, 2015.

[21] S. Kindt, J. Thienpondt, L. Becker, and N. Madhu. *Robustness of ad hoc microphone clustering using speaker embeddings: evaluation under realistic and challenging scenarios*. EURASIP Journal on Audio, Speech, and Music Processing, 2023(1):46, 2023.

[22] S. Kindt, M. Meeldijk, and N. Madhu. *Ad Hoc Distributed Microphones Clustering: A Comparative Analysis on Using Coherence and Signal-Specific Features*. In Speech Communication; 15th ITG Conference, pages 11–15. VDE, 2023.

[23] D. Lee and H. S. Seung. *Algorithms for non-negative matrix factorization*. Advances in neural information processing systems, 13, 2000.

[24] S. Rickard and O. Yilmaz. *On the approximate W-disjoint orthogonality of speech*. In 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), volume 1, pages I–529. IEEE, 2002.

[25] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen. *Permutation invariant training of deep models for speaker-independent multi-talker speech separation*. In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 241–245. IEEE, 2017.

[26] J. Chen, Q. Mao, and D. Liu. *Dual-Path Transformer Network: Direct Context-Aware Modeling for End-to-End Monaural Speech Separation*. In Proc. Interspeech 2020, pages 2642–2646, 2020. doi:10.21437/Interspeech.2020-2205.

[27] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe. *Deep clustering: Discriminative embeddings for segmentation and separation*. In 2016 IEEE international conference on acoustics, speech and signal processing (ICASSP), pages 31–35. IEEE, 2016.

[28] D. Diaz-Guerra, A. Miguel, and J. R. Beltran. *gpuRIR: A python library for room impulse response simulation with GPU acceleration*. Multimedia Tools and Applications, 80:5653–5671, 2021.

[29] R. Glitza, L. Becker, A. Nelus, and R. Martin. *Database of Simulated Room Impulse Responses for Acoustic Sensor Networks Deployed in Complex Multi-Source Acoustic Environments.* In EUSIPCO, 2023.

[30] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. *Librispeech: an asr corpus based on public domain audio books*. In 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP), pages 5206–5210. IEEE, 2015.

[31] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey. *SDR–half-baked or well done?* In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 626–630. IEEE, 2019.

[32] 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., volume 2, pages 749–752, 2001.

[33] 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, pages 4214–4217, 2010.

8

Efficient, Cluster-informed, Deep Speech Separation with Cross-cluster Information in Ad-hoc Wireless Acoustic Sensor Networks

This chapter follows up on the separation network of Chapter 7. In that chapter, the clusters are all processed independently. However, the other cluster contains useful information on the interferer that needs to be suppressed. Therefore, cross-cluster information exchange is added in this chapter. Additionally, increasing the temporal resolution of the DNN improves its performance. However, this leads to more computation complexity. Therefore, as second contribution, an efficient way of reducing this computational burden will be presented.

⋆ ⋆ ⋆

Stijn Kindt*, Jihyun Kim*, Hong-Goo Kang, Nilesh Madhu

Accepted to the 18th International Workshop on Acoustic Signal Enhancement (IWAENC 2024).

Abstract Environments with ad hoc distributed microphones, characterised by unpredictable locations and varying numbers, pose a significant challenge to most

* Equal contributions, shared first author

conventional deep-learning based speech enhancement and separation models. Transform-Average-Concatenate (TAC) layers in combination with dual-path transformer (DPT) models have been proposed for speech separation with flexible array configurations. For widely distributed microphone setups, we previously showed that blindly clustering microphones around target sound sources and processing each cluster separately yields good separation. In this work, we propose to further improve the output signal quality by exploiting *inter*-cluster information by suitable exchange between clusters using cross-attention transformers in the DPTs. Additionally, we introduce an efficient TAC, that lets us increase the temporal resolution and performance while keeping the computational complexity in check. Experiments in realistically simulated scenarios show increased separation quality by 1.1 dB SI-SDR, 0.02 improvement in STOI and 0.2 increase in PESQ, with significance at the median level.

8.1 Introduction

Spatial information is commonly exploited in multi-channel speech separation. Compact microphone arrays exploit the time difference of arrival or phase differences between the microphone signals for spatial filtering [1]. In Wireless Acoustic Sensor Networks (WASNs) [2], individual microphones or arrays are widely distributed throughout the environment, thereby providing supplementary spatial information. The microphones may be dominated by different sound sources and unwanted (background) noise components. This paper focuses on *ad hoc WASNs*, where the distributed microphones lack any predefined structure, and the structure is also, in general, unknown. Examples of such networks might include smartphones and laptops positioned on a meeting table, or they could be a combination of personal audio devices like earbuds, headsets, or hearing aids, connected with smart home devices in a living area.

While ad hoc WASNs offer extensive spatial coverage, they also present distinctive challenges not found in compact microphone arrays. Firstly, the microphone locations are not predetermined and may vary dynamically (*e.g.* the hearables moving with the speakers). Therefore the separation method should be independent of the microphone configuration. Secondly, the number of microphones is unknown and can also be time varying (*e.g.* the number of laptops on the conferencing table is not fixed and can increase or decrease as participants arrive or leave), further increasing the need for configuration-agnostic solutions. Thirdly, in WASNs, computational power is often limited, requiring smaller and computationally more efficient models. On top of this, other challenges for ad hoc distributed microphones include asynchronous sampling, leading to sample rate offsets (SROs) and sample time offsets (STOs), as well as significant differences in microphone characteristics (*e.g.*frequency response and directivity). We note that the robust solutions to

the latter challenges have been previously proposed [3–5], and focus, here, on the overarching problem of efficient array-agnostic separation in the absence of these issues.

However, most array-agnostic methods, e.g. [6–9], are tailored for *compact* ad hoc microphone arrays. Therefore, in [10] we exploited the assumption that microphones are dominated by different sources and proposed to cluster the microphones around the sources of interest. Using these clustered microphones as inputs, conventional array agnostic models can be applied to extract the underlying sound source of that cluster: dual-path transformers (DPTs) [11] process the spectral content for each microphone independently, while Transform-Average-Concatenate (TAC) layers [6] aggregate information across microphones in an array agnostic fashion.

The cluster informed deep network of [10] maximally exploits the information within the cluster. However, sharing information across clusters has proven invaluable in classical cluster-informed separation algorithms [12, 13]. Therefore, in this work, we evaluate the use of transformers with cross-cluster attention to replace the transformers of some DPTs. Similar ideas have been proven effective in music source separation, *e.g.*cross-instrument attention in [14] or cross-domain attention in [15].

Additionally, typically a model's efficacy can be improved by increasing the temporal resolution of the input. However, this increases the computational complexity drastically. Therefore, we propose to then compress the long-term information using strided convolutional layers in the most computationally demanding parts of the network. Together with a depthwise separable convolutional layer, the increased temporal resolution is performed in a efficient manner, while still ensuring improved performance.

Details about the method are in Section 8.2. Training setup and evaluation is discussed in Section 8.3 and conclusions are drawn in Section 8.4.

8.2 Methods

Figure 8.1 shows the general structure of the method as proposed in [10]. Initially, the microphones are clustered (top of the figure). We generate one cluster for each speaker, and one background cluster. The clustering algorithm additionally identifies the reference microphone for each speaker. Section 8.2.1 gives a brief description on clustering.

Following clustering, the array-agnostic separation network is employed to process all the microphones within the cluster. [10] showed the great efficacy of only utilising these microphones compared to all available microphones. Additionally, the use of the reference microphone to apply the mask and selecting the reference latent features after TAC has been found to significantly enhance performance.

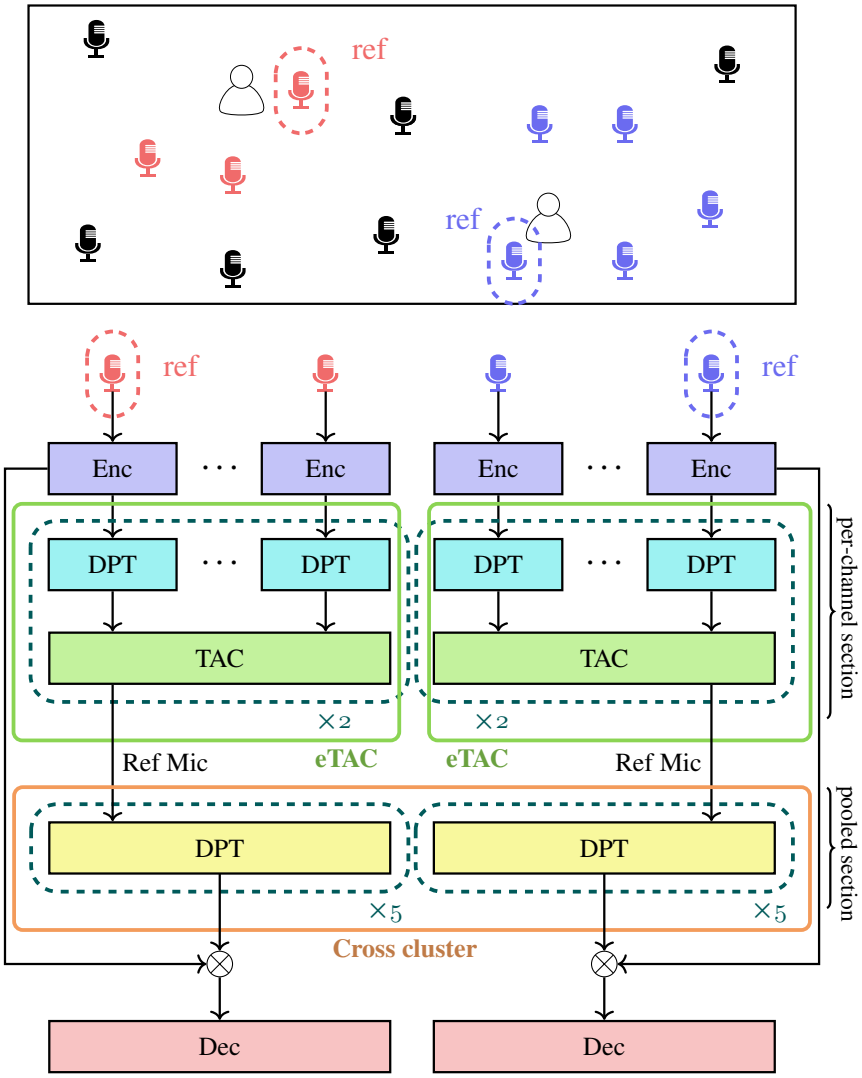


Figure 8.1: Scheme of the cluster informed deep separation network. The clustering is shown on top, with the separation method below. We propose to enhance the pooled DPT (yellow DPT) blocks with Cross cluster information and to change the per channel DPT+TAC blocks with eTAC blocks, depicted in Figure 8.2 and Figure 8.3 respectively.

More details will be given in Section 8.2.2.

The figure also indicated our proposed changes to the architecture. The cross-cluster attention DPTs (orange block) will be discussed in Section 8.2.3, while the proposed computationally efficient TAC layer (eTAC) (green block) is discussed in Section 8.2.4.

8.2.1 Clustering Algorithm

Numerous clustering methods for ad-hoc distributed microphones have been previously proposed [16–18]. Based on the comparative study in [19], we choose the full bandwidth coherence based method outlined in [17]. This method computes the magnitude-squared coherence between all possible combinations of M microphones, storing the results in the coherence matrix $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{M \times M}$. Next, non-negative matrix factorization (NMF) [20] is employed to decompose $\boldsymbol{\mathcal{C}}$, producing a clustering matrix $\boldsymbol{B} \in \mathbb{R}^{M \times C}$. The elements of this matrix, also called the fuzzy membership values (FMVs), indicate the degree of belonging of a microphone for a given cluster: if $\boldsymbol{B}_{mc}$ is high, microphone m strongly correlates to other microphones of cluster c. The final (hard) clustering is decided by assigning each microphone to the cluster with the highest FMV. Additionally, each cluster can select a reference microphone based on the microphone with the highest FMV towards that cluster. Both, the hard clusters and reference microphones are used for *informed* speech separation.

8.2.2 Cluster-informed Deep Separation Network

Encoder. The encoder comprises a single layer 1D convolutional layer applied to the time domain signals of each microphone, with F filters, ReLU activation function, a kernel size of Q and a stride of $Q/2$. At the end of the encoder, the segmentation strategy of dual-path RNNs is performed [7]. This segmentation strategy divides the encoder output into overlapping chunks of size K and 50% overlap, yielding a tensor that encodes short-time information along one axis and long-term information along the other:

$$\text{Enc}(x_{m,c}) = \text{Segment}(\text{ReLU}(\text{Conv1D}(x_{m,c})), \tag{8.1}$$

where x are the time-domain input signals, and m, c indicate the microphone and cluster indices respectively. Normalisation of the input is performed with layer norm.

Separator. The separator takes inspiration from the Vararray architecture [8] and can be divided into a per-channel section and a pooled section. The per-channel section consists of a DPT block [11], interleaved with TAC layers. The DPT block processes information at each microphone channel independently with Transformers [21] along the short- and long-term dimensions, called LongTermTrans and

ShortTermTrans respectively:

$$\textbf{DPT}(h_{m,c}) = \text{LongTermTrans}(\text{ShortTermTrans}(h_{m,c})), \tag{8.2}$$

while TAC blocks [6] exchange information across microphones:

$$\textbf{TAC}(h_{m,c}) = \text{Concat}(\text{FC}_1(h_{m,c}), \frac{1}{M_c}\sum_{m\prime=0}^{M_c-1}\text{FC}_2(h_{m\prime,c})). \tag{8.3}$$

h represents the input feature of that layer, M_c is the number of microphones for that cluster and FC is a linear transformation followed by ReLU activation. For each of TAC's output channels, half of the output comes from that channel, while the other half is averaged over all other channels. This process is repeated twice, after which pooling is applied to reduce computational complexity. In [10], it was shown that selecting the features corresponding with the reference microphones is the optimal pooling strategy.

In the pooled section, a series of five DPTs is applied to the pooled channel. The output of the separator is a mask in latent space, which is applied to the encoder embedding of the reference microphone of each cluster.

Decoder. The decoder reverses the operations of the encoder on the masked embeddings for each cluster:

$$\hat{y}_c = \text{Dec}(h_c) = \text{DeConv1D}(\text{Overlap-Add}(h_c)). \tag{8.4}$$

It serves as the network output, representing the clean speech estimate $\hat{y}_c$ for each cluster c.

8.2.3 Cross-cluster attention

Classical separation methods [12, 13] exploit cross cluster information to be able to suppress the unwanted components. However, in our recent cluster informed separator [10], no such information exchange was done. The network relies solely on the assumption that all microphones of a cluster are dominated by the same source, to extract that source. We propose to incorporate cross cluster information exchange in the separator so that it has a clearer reference of the interfering source as well. The information exchange is performed in the pooled section of the deep separation network, as illustrated in Figure 8.2. Here, every other DPT block employs cross-attention instead of self-attention [21], where the keys and values constitute the inter-cluster information, for both the short-term and long-term transformer layers of the DPT blocks. Interleaving cross and self attention DPT blocks ensures that each cluster can appropriately process the information received from the other cluster. Note that replacing the self-attention blocks with the cross-attention blocks maintains the same network complexity as the baseline.

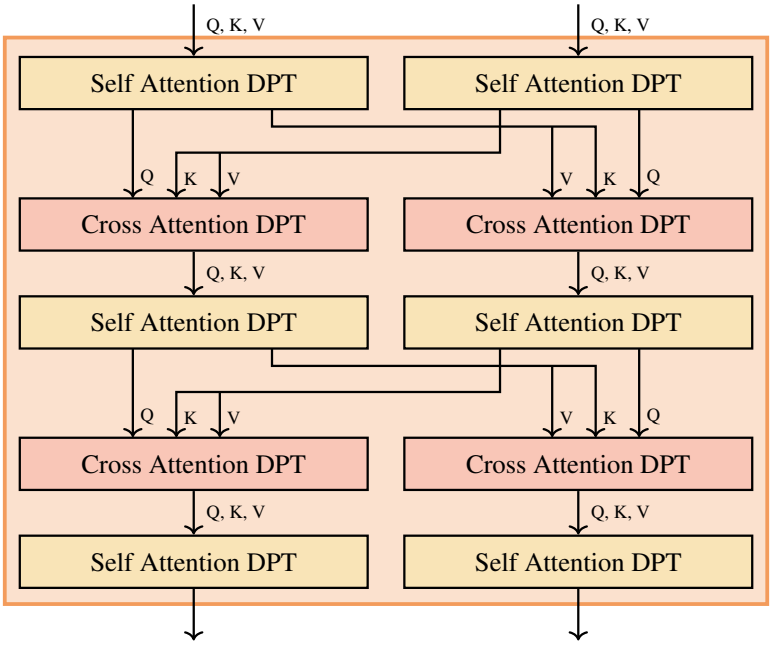


Figure 8.2: Scheme of the proposed cross cluster information exchange. Q, K and V indicate where the queries, keys and values come from in the transformer's attention heads.

8.2.4 Efficient TAC

Increasing the temporal resolution of the encoder, and thus also the temporal resolution of the subsequent layers, reasonably leads to increased performance. By lowering the stride size, the encoder can capture finer details of the speech signal. However, this enlarges the number of hidden features throughout the network, resulting in an increased computational cost. This is not desirable, certainly in WASNs. The computational complexity is mainly situated in the per-channel section of the network, and increases rapidly with increasing number of clustered microphones. To still raise temporal precision, we propose an efficient TAC (eTAC), which compacts feature representations in the per-channel section.

As illustrated in Figure 8.3, the eTAC layer employs the depth-wise separable (DWS) convolution layer [22] and a squeeze convolutional layer to compress the feature dimensions generated by the encoder. These consist of depth-wise (**Depth Conv**) convolutions, followed by a point-wise convolution (**Point Conv**), which is a more efficient convolutional method, and aids in combining the information along the long-term axis effectively. The squeeze convolutional layers (**Squeeze Conv**) use a 4×3 kernel and a 2×1 stride to compress the long-term temporal dimensions.

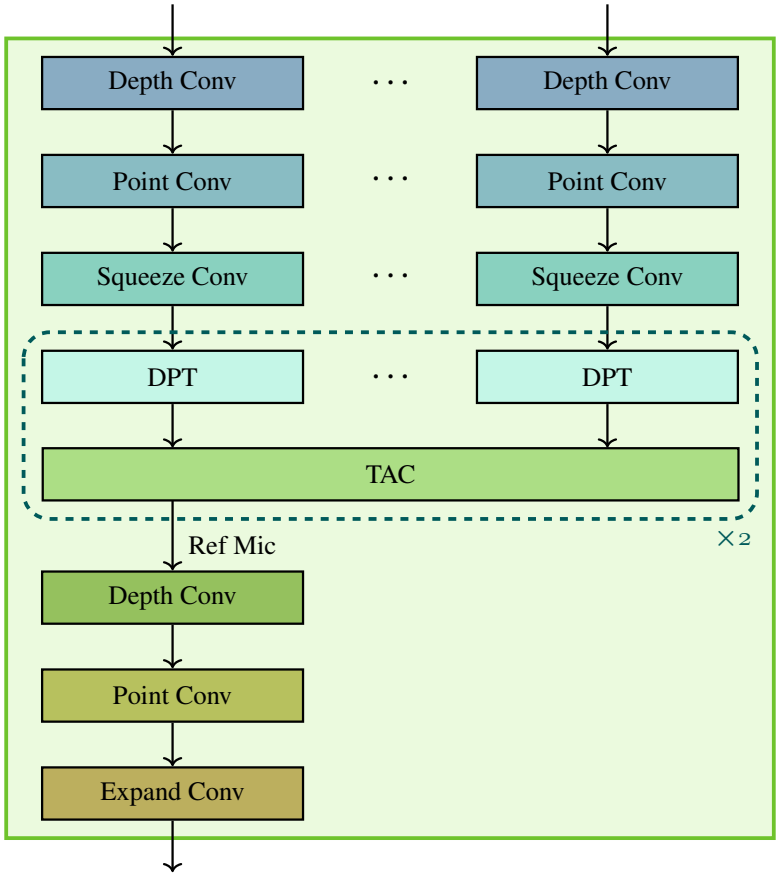


Figure 8.3: Scheme of the proposed efficient TAC block.

After all the information across the different microphones is processed with the DPT and TAC layers, and the pooling operation is applied, the feature size can once again be increased. The pooled DPT blocks are far less computationally demanding, since they only need to process one (microphone) channel per cluster. The feature size is increased to the original size by another DWS convolution layer followed by a deconvolutional block (**Expand Conv**), with the same parameters as the squeeze convolutional layer. This ensures that the last blocks can fully utilise the increased temporal resolution.

Note that eTAC could also be deployed without increasing the temporal precision. This would reduce the computational complexity of the system. We will also evaluate to what extent the performance is sacrificed with eTAC.

8.3 Evaluation

8.3.1 Training paradigm

The training setup closely resembles the one described in [10]. However, it is important to note that although the proposed method exploits clustering information, no actual clustering is conducted during training. Executing clustering during training would consume valuable time, as clustering needs to be performed for each example. Instead, room impulse responses (RIRs) are simulated with microphones positioned to resemble those of clustered microphones. To achieve this, two speaker positions are randomly generated within the room. To ensure they are sufficiently spaced apart, these positions are generated in different halves of the room. Following this, three microphones are simulated within the critical distance of each speaker, while four microphones are generated within a $2m \times 2m$ square centred around each speaker. Various room sizes and reverberation times are simulated, along with different source and microphone positions, resulting in a total of 10,080 sets of RIRs.

During training, a random number of microphones between 3 and 7 are selected for each cluster to train independently to the number and geometry of microphones. This can be a different random number for each cluster. The microphones within the critical distance are always chosen, mimicking the distribution of microphone positions selected by the clustering algorithm. These RIRs are simulated using the shoebox image source model of gpuRIR [23]. Subsequently, the RIRs are convolved with a random sample from the WSJ0-2mix dataset [24] and white noise is added at signal-to-noise ratios (SNRs) uniformly sampled between 0 dB and 20 dB.

8.3.2 Experimental setup

For the evaluation, except if specifically mentioned, the following parameters are chosen. The sampling rate is $f_s = 16$kHz, the number of Encoder filters $F = 64$, kernel size is $Q = 16$ samples with a stride of 50%. We use a segment size $K = 250$ for the segmentation strategy of dual-path processing. Each transformer has a dimension of $F = 64$ and uses 4 attention heads. The training criterion we used is the Scale Invariant Signal to Distortion Ratio (SI-SDR) [25] loss, aiming to minimise the following loss function:

$$\mathcal{L}_{SI-SDR} = -10 \log_{10} \frac{||\alpha s||^2}{||\alpha s - \hat{s}||^2}, \tag{8.5}$$

where $\alpha = (\hat{s}^T \cdot s)/||s||^2$ is a scaling factor, s is the clean speech and $\hat{s}$ is the estimated speech signal. The scaling factor makes it so the output is independent of scale and eases the network's learning. We used the Adam optimiser and the

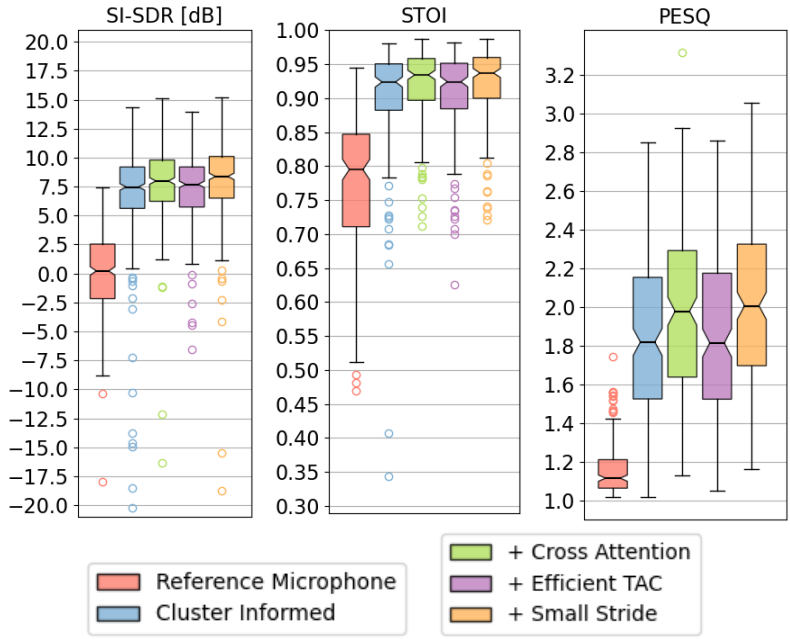


Figure 8.4: SI-SDR, STOI and PESQ for the proposed changes separation methods (higher is better). '+' indicates that the changes are accumulative w.r.t. the preceding method.

training process began with an initial learning rate of 0.125, with a strategy to halve the learning rate if the validation loss doesn't decrease for three epochs.

8.3.3 Evaluation scenarios

The evaluation scenarios are similar to the ones used in [18, 19], and different to the training data. Realistically simulated RIRs are obtained from the SINS dataset [26], which used a CATT model [27]. For each scenario, two speaker positions are chosen from the opposing halves of the room. Note that the speakers could still be close to each other if they are both sampled towards the middle of the room. Then 16 microphones are selected from all the possibilities within the dataset. It is made sure that at least 3 microphones are within the critical distance of each source. the Librispeech dataset [28] is used for dry speech signals, and white noise is added at each microphone. The noise level is the same at each microphone and is set to be 10dB SNR at a virtual microphone in the centre of the room. The clusters are decided by the clustering algorithm described in Section 8.2.1. 200 scenarios are run, and metrics are calculated for each source, giving 400 examples in total.

To assess the performance, we employed three objective metrics: Scale Invariant Signal to Distortion Ratio (SI-SDR) [25], Perceptual Evaluation of Speech

Table 8.1: Ablation study of the number of trainable parameters and Giga Multiply Accumulate Operation per second (GMACs). The GMACs are split in a fixed contribution, in the case a cluster only contains one microphone and a per microphone cost, which is the added complexity per extra microphone in the cluster.

Stride size ($Q/2$)	**Cross Cluster**	**eTAC**	**Params.**	**GMACs**	
				Singe mic	Per additional mic
8	✗	✗	4.761 M	27.567 G	7.894 G
4	✗	✗	4.760 M	52.021 G	14.907 G
8	✓	✗	4.761 M	27.567 G	7.894 G
4	✓	✗	4.760 M	52.021 G	14.907 G
8	✓	✓	4.869 M	24.339 G	4.212 G
4	✓	✓	4.868 M	45.925 G	7.953 G

Quality (PESQ) [29], and Short-Time Objective Intelligibility (STOI) [30]

8.3.4 Separation quality

Figure 8.4 shows the separation metrics of the different methods. The unprocessed reference microphone (from the clustering method) is given as a quality indication for the input signal. The cluster informed model of [10] is also plotted, to which the proposed methods are cumulatively added. First, we note that changing two self attention DPT blocks with their cross attention counterparts is beneficial for the system. The notched box plots indicate that all instrumental metrics increase with significance at the median level, demonstrating the benefit of cross-cluster informed separation.

Next, we notice that directly adding the eTAC on top of cross cluster attention *degrades* the performance, to slightly lower instrumental metrics than the baseline. This is due to the loss of information while at a stage where information exchange happens across microphones within the cluster. To compensate for this, the time resolution is increased by lowering the stride of the encoder to 4 instead of 8. This leads to a performance, based on the instrumental metrics, that slightly outperforms the cluster informed and cross cluster attention methods, showing that lowering encoder stride can capture more information from the input. Audio examples are available at https://aspire.ugent.be/demos/IWAENC2024SK/, to better appreciate the cumulative improvements obtained.

8.3.5 Model efficiency

The stride size of the encoder directly influences the size of the hidden features being processed, significantly impacting the computational complexity. A larger stride results in more compressed hidden features, enabling efficient processing but at the cost of potential information loss, leading to a trade-off with separation performance. As shown in Table 8.1, using eTAC allows for the use of smaller

stride sizes without substantially increasing the computational complexity. Compared to other baseline models that use the same stride size, eTAC significantly reduces computational complexity. The computational complexity per additional microphone, when using a lower stride size with eTAC, is comparable to that using a larger stride without eTAC (compare rows 3,4, and 6).

8.4 Conclusions

In this paper, we addressed the cluster-informed separation of two speakers in ad hoc distributed microphone environments. Two novel proposals were made: first, adding cross cluster attention and second, using an efficient TAC (eTAC) layer, which enables a higher temporal resolution while keeping the computational complexity in check. Both proposed improvements demonstrated benefits in instrumental metrics. The cross cluster attention improved the performance the most, while eTAC provided marginal improvements. Nevertheless, it may be seen as a flexible method to trade-off computational complexity and performance.

Acknowledgment

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

References

[1] A. Bohlender, A. Spriet, W. Tirry, and N. Madhu. *Spatially Selective Speaker Separation Using a DNN With a Location Dependent Feature Extraction*. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2023.

[2] A. Bertrand. *Applications and trends in wireless acoustic sensor networks: A signal processing perspective*. In 2011 18th IEEE symposium on communications and vehicular technology in the Benelux (SCVT), pages 1–6. IEEE, 2011.

[3] T. Gburrek, J. Schmalenstroeer, and R. Haeb-Umbach. *On Synchronization of Wireless Acoustic Sensor Networks in the Presence of Time-Varying Sampling Rate Offsets and Speaker Changes*. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 916–920, 2022. doi:10.1109/ICASSP43922.2022.9746284.

[4] A. Chinaev, N. Knaepper, and G. Enzner. *Long-Term Synchronization of Wireless Acoustic Sensor Networks with Nonpersistent Acoustic Activity Using Coherence State*. In ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1–5. IEEE, 2023.

[5] R. Wang, Z. Chen, and F. Yin. *Distributed frequency response calibration based on consensus strategy in microphone array*. IEEE Transactions on Instrumentation and Measurement, 70:1–12, 2021.

[6] Y. Luo, Z. Chen, N. Mesgarani, and T. Yoshioka. *End-to-end microphone permutation and number invariant multi-channel speech separation*. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6394–6398. IEEE, 2020.

[7] Y. Luo, Z. Chen, and T. Yoshioka. *Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation*. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 46–50. IEEE, 2020.

[8] T. Yoshioka, X. Wang, D. Wang, M. Tang, Z. Zhu, Z. Chen, and N. Kanda. *VarArray: Array-geometry-agnostic continuous speech separation*. In ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6027–6031. IEEE, 2022.

[9] H. Taherian, S. E. Eskimez, T. Yoshioka, H. Wang, Z. Chen, and X. Huang. *One model to enhance them all: array geometry agnostic multi-channel personalized speech enhancement*. In ICASSP 2022-2022 IEEE International

Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 271–275. IEEE, 2022.

[10] J. Kim, S. Kindt, N. Madhu, and H.-G. Kang. *Enhanced Deep Speech Separation in Clustered Ad Hoc Distributed Microphone Environments*. In Proc. Interspeech 2024, pages 1–5, 2024.

[11] J. Chen, Q. Mao, and D. Liu. *Dual-Path Transformer Network: Direct Context-Aware Modeling for End-to-End Monaural Speech Separation*. In Proc. Interspeech 2020, pages 2642–2646, 2020. doi:10.21437/Interspeech.2020-2205.

[12] S. Gergen, R. Martin, and N. Madhu. *Source separation by feature-based clustering of microphones in ad hoc arrays*. In 2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC), pages 530–534. IEEE, 2018.

[13] S. Gergen, R. Martin, and N. Madhu. *Source separation by fuzzy-membership value aware beamforming and masking in ad hoc arrays*. In Speech Communication; 13th ITG-Symposium, pages 1–5. VDE, 2018.

[14] J. Kim and H.-G. Kang. *Contrastive learning based deep latent masking for music source separation*. In Proceedings of INTERSPEECH, pages 3709–3713, 2023.

[15] S. Rouard, F. Massa, and A. Défossez. *Hybrid transformers for music source separation*. In ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1–5. IEEE, 2023.

[16] S. Gergen, A. Nagathil, and R. Martin. *Classification of reverberant audio signals using clustered ad hoc distributed microphones*. Signal Processing, 107:21–32, 2015.

[17] A. J. Muñoz-Montoro, P. Vera-Candeas, and M. G. Christensen. *A Coherence-based Clustering Method for Multichannel Speech Enhancement in Wireless Acoustic Sensor Networks*. In 2021 29th European Signal Processing Conference (EUSIPCO), pages 1130–1134. IEEE, 2021.

[18] S. Kindt, J. Thienpondt, L. Becker, and N. Madhu. *Robustness of ad hoc microphone clustering using speaker embeddings: evaluation under realistic and challenging scenarios*. EURASIP Journal on Audio, Speech, and Music Processing, 2023(1):46, 2023.

[19] S. Kindt, M. Meeldijk, and N. Madhu. *Ad Hoc Distributed Microphones Clustering: A Comparative Analysis on Using Coherence and Signal-Specific*

Features. In Speech Communication; 15th ITG Conference, pages 11–15. VDE, 2023.

[20] D. Lee and H. S. Seung. *Algorithms for Non-negative Matrix Factorization*. In T. Leen, T. Dietterich, and V. Tresp, editors, Advances in Neural Information Processing Systems, volume 13. MIT Press, 2000. Available from: https://proceedings.neurips.cc/paper/2000/file/f9d1152547c0bde01830b7e8bd60024c-Paper.pdf.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. *Attention is all you need*. Advances in neural information processing systems, 30, 2017.

[22] F. Chollet. *Xception: Deep learning with depthwise separable convolutions*. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1251–1258, 2017.

[23] D. Diaz-Guerra, A. Miguel, and J. R. Beltran. *gpuRIR: A python library for room impulse response simulation with GPU acceleration*. Multimedia Tools and Applications, 80:5653–5671, 2021.

[24] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe. *Deep clustering: Discriminative embeddings for segmentation and separation*. In 2016 IEEE international conference on acoustics, speech and signal processing (ICASSP), pages 31–35. IEEE, 2016.

[25] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey. *SDR–half-baked or well done?* In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 626–630. IEEE, 2019.

[26] R. Glitza, L. Becker, A. Nelus, and R. Martin. *Database of Simulated Room Impulse Responses for Acoustic Sensor Networks Deployed in Complex Multi-Source Acoustic Environments.* In EUSIPCO, 2023.

[27] B.-I. Dalenbäck. *CATT-Acoustic™ v9.1 powered by TUCT™ v2*. https://www.catt.se/, 2019. Accessed: 2024-07-12.

[28] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. *Librispeech: an asr corpus based on public domain audio books*. In 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP), pages 5206–5210. IEEE, 2015.

[29] 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., volume 2, pages 749–752, 2001.

[30] 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, pages 4214–4217, 2010.

9 Conclusions

9.1 Research contributions

Chapter 3: Location informed deep speech separation.

At first, the thesis described how location based input (LBI) separation DNN with a single microphone array in Chapter 3. There, the addition of locational information helped for separating closely spaced sources. However, for widely separated sources, this specific method of incorporating location information did not improve the separation performance. This could be because the location based outputs (LBO) system used as baseline [1, 2], which is location aware by generating a mask for each direction, is already discriminatory.

Two types of LBI features were tested, the hand crafted 'expected phase difference' and the learned representation by the deep neural network, generated from a multi-hot vector. From the two, the DNN learned representation worked best. This is likely because the expected phase difference features are based on the simplifying assumption of plane wave propagation, while the DNN can learn a more general representation.

Chapter 4: Source localisation with distributed microphone arrays.

The second work moved towards distributed microphone arrays and proposed a method to combine the arrays for more accurate localisation. These accurate localisations are crucial for methods like LDI since they often assume perfect

location information during training. The challenges associated with distributed microphone array processing were considered while designing the DNN. the challenges were specifically: (i) the microphones between two arrays are far apart, leading to spatial aliasing and (ii) the clocks of the two arrays cannot be assumed synchronous.

The most straightforward method that fulfils these requirements is to triangulate the individual array's outputs. However, the chapter showed that the performance can be improved by letting the networks co-operate. Note that sharing input audio would violate the two main challenges: aliasing would occur and clock offsets would make it hard for the network to learn anything useful. Two methods were tested, one mixed narrowband network features, while the other shared broadband features. At first, exchanging more information between the arrays would seem advantageous, but the NM-CLA performed worse than the BM-CLA. This could be assigned to the aliasing effect that could still be present when mixing narrowband information across large distances. Nevertheless, both methods are inherently robust against sample rate offsets since only signals of the same array, and thus the same clock, are directly combined.

Chapter 5: Speaker embedding based microphone clustering.

Up till this point of the thesis, the locations of the microphones were assumed known with respect to each other. However, it is also interesting to perform localisation and separation with ad-hoc distributed microphones. In this work, speaker embedding features were proposed to perform soft clustering. This indicates the relative location of the microphones in the sense that microphones close to the same source will be clustered together.

The newly proposed speaker embeddings were taken from the pre-trained speaker verification network ECAPA-TDNN. They showed an improvement over the other speaker specific feature Mod-MFCC in terms of Direct-to-Reverberant Ratio (DRR), Direct-to-Reverberant-Interference-and-Noise Ratio (DINRR) and separation quality. Here, we also noticed it is advantageous to select an appropriate distance metric in the clustering algorithm. Cosine distance, which is what the speaker verification networks are optimised for, is a more optimal choice than Euclidean distance. This also holds for the Mod-MFCC features, since these features should be scale-invariant. Evaluations on challenging scenarios of closely spaced sources and shorter time frames on which the clustering features are based also showed that speaker verification embeddings are the best choice. Additionally, the speaker verification features could be used for target speaker extraction, showcased with an empirical proof-of-concept example. Lastly, these examples were carried out on more realistic data than previous works have showcased.

Chapter 6: Coherence vs Signal-Specific clustering of distributed microphones.

This work continues on the evaluation of the best feature to cluster microphones on. Coherence based features are used and compared to the speaker specific features of Chapter 5. Here, the two main hypotheses were incorrect. The first hypothesis was that coherence based clustering would perform worse than speaker specific features. However, the coherence based features are equal to or even very slightly better than the speaker specific features, and selected a better reference microphone. The second wrong hypothesis was about the use of lossy codecs. Here the assumption that the non-linear distortions from the codec would eliminate the usefulness of the coherence metric. However, as it turns out, the subband codec preserves enough coherence at the lower frequencies, and the clustering method with NMF only cares about relative coherence between microphone pairs.

Chapter 7: Cluster informed deep speech separation.

The last two works focused on DNN based separation in ad-hoc distributed microphone scenarios. In this specific chapter, the idea is to utilise the clusters given from the previous chapters to inform the DNN structure. Firstly, a time domain DNN was described that is array agnostic and based on the Vararray structure [3]. Array agnostics is an important property to handle the ad-hoc disturbed microphone. However, these are typically designed for compact ad-hoc microphone arrays. In this chapter, therefore, a comparison between inputting all microphones to such a network – the unclustered version – and a version where only the microphone clustered around a source – the clustered version – was performed. Since the clustered microphones should all be relatively close to each other, the compact assumption should hold.

The cluster information has proven essential for the separation to perform well. The unclustered version cannot deal with all the microphones being dominated by different sources. In fact, since each cluster has its own dominant source, the target of the network changed from source separation to running two networks on each cluster to perform source extraction. A comparison with a single-channel extraction DNN, where the chosen channel is the reference microphone – the best microphone according to the clustering. Through this comparison, it is clear that exploiting all microphones for each cluster is beneficial. Nevertheless, only processing the reference microphone also delivers decent results. It is therefore unsurprising that promoting the reference microphone, in the clustered DNN improves the performance further.

Chapter 8: Efficient and cross-cluster deep speech separation.

The last contributions contain extending the cluster informed DNN separation in ad-hoc distributed microphone setups in two ways. The first was to introduce cross cluster attention, which aided in suppressing the interfering source, associated with the other cluster. The second contribution was to introduce a squeeze

convolution (strided convolution), that reduced the feature dimension in order to free up computational power. Redistributing that power to increase the temporal resolution provided a slight improvement.

9.2 Backward glance: potential research branches

Reflecting on the work presented over the years offers some new perspectives and highlights potential alternative design choices that might be made today. Additionally, I want to revisit some interesting branching research directions that were left unexplored due to my focus on other aspects. These abandoned branches could still lead to interesting research results and might serve as inspiration for anyone interested in pursuing these specific research topics. Furthermore, I would like to acknowledge the contributions of others who have in the meantime conducted research in some of these branches.

9.2.1 Location informed separation (Chapter 3)

Alternative location informed formulations

Compact microphone array processing is a popular topic within the audio enhancement community. Therefore, others have since this publication made iterations and improvements to this work. For instance, as an alternative to incorporating the location information of the speaker at the input of the network, [4, 5] used location information to initialise their recurrent neural network. [2] on the other hand proposed to train specific feature extraction layers for each DOA section, termed Location Dependent Feature Extraction (LDE). This proved to be a superior method of incorporating location information into deep separation networks. Alternatively, [6] proposed to give location information but relax its definition. The network receives a look direction and is trained with the *goal* of extracting the speaker that is closest to that look direction. Providing this look direction does help to separate the sources over providing no information at all.

Robustness against location errors

In this work ([7]), it is assumed that the speaker locations are perfectly known. No estimation errors are assumed here. This is however not realistic, and some localisation errors are to be expected. An evaluation of the allowed deviations to the true source positions should be carried out. For example, in [5], a DOA error beyond 2 degrees shows significant degradation in the resulting speech quality. Note that they utilised finer angular sectors for their one-hot representation: two degree sectors compared to 5 degree sectors in [7]. Therefore, a two degree error would likely still correspond to the same multi-hot vector. Nevertheless, the conclusion

of degrading performance as the localisation accuracy goes down should still hold. As one solution, [5] added inaccurate location information in the training loop, which did make the model more robust against localisation errors. However, the performance dropped when the source was perfectly localised. Similarly, although [6] provides a look direction which by definition should not be accurate, having precise location information is still superior.

Besides the problem of errors in the location estimates, correctly locating closely spaced sources is, similar to separating them, very difficult. The localisation DNN could be confused with the number of speakers active in that general direction, and only identifying one source while two are active. This problem cannot straightforwardly be solved with one microphone array. Both problems highlight the advantage of having multiple microphone arrays. For one microphone array, two speakers might be very close in their DOA, for an array positioned elsewhere the sources might be wider spaced. Additionally, multiple arrays can combine to provide a more accurate localisation.

9.2.2 Source localisation with distributed microphone arrays (Chapter 4)

Bandwidth consideration

The broadband mixing co-operative localisation architecture (BM-CLA) came out on top compared to the narrowband mixing CLA (NM-CLA) method, which is also beneficial in terms of bandwidth requirements to send the features to a central node. However, it might be possible to reduce the bandwidth even further, since this was not the focus of that chapter. In [8], we proposed to add a convolutional encoded and decoder to compress and decompress the features respectively. This way, the bandwidth is further reduced in three major ways. firstly, the data can be compressed along the time dimensions with strided convolutional layers. secondly, the feature dimension can be reduced. And, thirdly, the features can be quantised. That work focused on localisation with hearing aids, where the two earpieces were regarded as two distributed arrays. An evaluation of the free field arrays of Chapter 4 would further be interesting. Additionally, the quantisation scheme was simple yet very effective. More quantisation might be possible through Vector Quantised-Variational AutoEncoder (VQ-VAE) [9].

Input structure

The work in Chapter 4 had the localisation networks of [10] as a basis, which extended the network of [11] with a recurrent layer. However, the original design ([11]) was specific for linear microphone arrays. The microphones in a linear microphone array, as the name suggests, are positioned at equal distances from

each other in a straight line. This ensures that the relation between two neighbouring microphones is translational invariant. Therefore, a convolutional layer that goes along the microphone dimensions in order to combine them makes perfect sense. However, in the work done in [12], two square microphone array was used, making the relation between each two neighbouring microphones not the same anymore. Sharing the weights to combine neighbouring microphones – which happens with convolutional layers – could therefore be suboptimal. Experiments where the weights are no longer coupled, but instead independent for each microphone pair, could indicate whether the DNN could learn more relevant features. This would increase the number of trainable parameters but, perhaps more importantly, keep the computational complexity the same and potentially increase performance.

Amplitude information

I also wanted to note this model currently does not incorporate amplitude information. Only the phases of the microphone spectrum are used as input features, which is again in line with the prior work it is based on [11]. For compact microphone arrays, this makes sense since the amplitude differences between the microphones are rather small because they are so close to each other. In contrast, the distance between the microphone arrays is quite large, making the amplitude difference between the arrays could be of particular interest in determining the exact location of the sources. Additionally, amplitude information could help indicate what time-frequency bins are dominated by speech, which would increase the performance in noisy conditions.

Geometry aware

Lastly, the location of the two arrays is assumed known with respect to each other. Although this allows for some flexibility in where these arrays are placed in the room and the size of that room, this does not account for ad-hoc distributed microphone scenarios. The system should be made geometry aware to avoid retraining for each new scenario. Triangulation approaches can directly be applied to these scenarios since the individual DOA estimates do not depend on the position of the other arrays, and the triangulation can take in new array locations to compute the 2D source position. However, it was shown that co-operative networks provide a higher accuracy.

There are recent works out there that use the microphone positions as inputs for the system. This way, the network could learn to deal with different array setups, while still exploiting the co-operative nature. For instance, in [13, 14], they use individually distributed microphones and use one hot encoded vectors, representing the microphone positions, as additional inputs. Note that this is a very similar input representation to how the source positions were encoded in Chapter 3, only in 3D/2D space instead of representing the DOAs. However, inputting the

position of the microphones/arrays in such a one hot vector fashion will lead to quantisation errors. And since the microphone signals depend in a non-linear manner on their location, this might lead to non ideal errors in the output. In contrast, [15] has proposed to use cartesian coordinates as input features, which do not suffer from this quantisation error. However, this work focuses on ad-hoc compact microphones and no experiments on distributed microphone arrays have been carried out. Therefore, this would be interesting to research in the future.

9.2.3 Clustering of ad-hoc distributed microphones (Chapters 5 and 6)

NMF speaker similarity matrix

In Chapter 6, a comparison of the coherence based features and speaker specific features are performed. However, not only the features differed, but also the clustering method. The speaker specific features utilised Fuzzy-C Means (FCM) clustering, while the coherence based clustering was based on Non-Negative Matrix Factorisation (NMF). As an alternative, the speaker embeddings can form a speaker similarity matrix, similar to the coherence matrix from [16]: The elements of the $M \times M$ matrix would then be cosine similarity of the different microphone pairs. This way, the influence of the clustering method can be evaluated directly. For instance, the FCM algorithm works by having the cluster centre as a weighted average of the embeddings (dominated mostly by the embeddings of that cluster). And since the reference microphone is the one that is closest to that cluster centre, it is closest to the average embedding. This does however not ensure that the best microphone is picked as the reference microphone. In contrast, the NMF based on the coherence features did select a better reference microphone more often [17]. Thus combining speaker embeddings with NMF can indicate whether this is a property of the coherence features or the clustering technique.

Combining coherence and speaker-specific features

Both features have their advantages. Firstly, the speaker embeddings have fixed bandwidth requirements, independent of the signal length over which it is computed. For the coherence based features, the bandwidth requirements are larger, even if the signals are encoded. Additionally, the speaker embeddings could be used to directly identify clusters of interest based on known target embeddings. In contrast, the coherence based method performs slightly better and the sent signals are useful for subsequent tasks like speech separation.

To combine the advantages of both, both clustering techniques could be used in sequence. For instance, in the separation of the cluster informed deep separation [18], only the microphone signals of the target cluster are needed. Therefore, the

embedding based clustering can be used to select the microphones in a bandwidth efficient manner, and with the target embedding in mind. However, the best cluster-informed separation utilises the reference microphone signal explicitly and expects this to be the best one. Therefore, coherence based clustering could be used to further identify the optimal reference microphone.

9.2.4 Clustering based deep separation (Chapters 7 and 8)

Number of microphones in each cluster

In this work [18], a comparison between a single channel separation framework and a framework utilising all microphones in the cluster was performed. The latter approach, which included all clustered microphones, demonstrated superior performance. However, it would also be valuable to investigate the contribution of each additional microphone. Others have performed similar evaluations, such as in [19], where the benefit of adding extra microphones to the Delay-and-Sum Beamformer (DSB) was found to be negligible beyond four or five. Lowering the number of microphones needed would reduce the computational complexity significantly.

Input representation

In the current separation framework, time domain signals are directly inputted to the neural network. Although, many frameworks use this effectively, a comparison with Time-Frequency (TF) input features from the Short Time Frourier transform (STFT) be interesting. The speakers are assumed approximately disjoint in the TF domain [20], making it a perfect representation for masking. Further, the TF masks are ideal for computing statistical beamformers like the MVDR [21, 22] and Multichannel Wiener Filter (MWF) [23]. Recently, [24] compared time domain and TF domain features for single channel deep dereverberation. There, TF representations performed slightly better. Therefore, future work should compare time domain inputs with TF inputs for clustered speech separation.

Real time on edge processing

Currently, the separation model is quite computationally demanding and uses non-causal information: the model computes its output on segments of 2 seconds. Both drawbacks render this system unusable for applications where the latency should be limited – *e.g.*direct playback of the enhanced signal – and on edge computation is required instead of cloud computation. Therefore, future research should focus on making the model causal by changing the non-causal long term transformer with its causal variant, or with a recurrent layer. Additionally, the computational complexity should be lowered. Many strategies should be compared

here in future work.

9.3 Future perspectives: move towards ad-hoc distributed microphone arrays

Looking back at what design choices would be made differently today and at alternate research branches is always useful. However, examining how the research contributions could be exploited in new applications is even more exciting. A multitude of different array configurations were considered during the thesis: from compact microphone array(s) to ad-hoc distributed individual microphones. Yet, phones, laptops and smart glasses often have access to more than one microphone, making them ad hoc distributed microphone *arrays*. Additionally, multiple dedicated multichannel devices could be set up in meeting rooms or living areas. This is particularly important for large, reverberant rooms to ensure better coverage of the space, or in scenarios where many speakers are active simultaneously, and the different devices can be strategically positioned to cover the various sources.

Since the array positions, nor the source positions are known in these ad-hoc setups, a straightforward method of combining the information at the different arrays is not available. However, a couple of strategies could be deployed. Firstly, each microphone array could blindly beamform in all directions. For instance, each microphone array can steer its beams in four different look directions, extracting sources from these directions only. This will yield a set of virtual microphones, containing either a speaker or mostly noise. Without the location information, it is impossible to directly know which beams from different arrays capture the same speaker. Clustering, as proposed in Chapters 5 and 6, could therefore be used to cluster the virtual microphone signals, instead of the physical microphone signals. This has already been done in [25, 26]. However, the beamformed signals have not yet been combined for better separation performance. Here, the works of Chapters 7 and 8 could be used, where the inputs are no longer the raw microphone signals, but the virtual beamformed signals that are clustered together.

Another aspect of ad-hoc distributed arrays is that it is possible to do joint localisation of the array positions and source positions. This is typically called geometry calibration and works by iteratively estimating the arrays' and sources' positions [27]. In the fully blind approach, each array performs DOA estimates and distance estimated on its own. Here, clustering could come in handy, informing which estimates belong to each other. Still, the measurements could be noise. Here is where the co-operative localisation architectures of Chapter 4, if made geometry aware, could come in handy, for an improved localisation.

After precisely localising the sources, the location informed separation of Chapter 3 could be utilised for a better source extraction than blindly beamforming

towards all directions as previously mentioned. This could in turn improve the cluster based separation, that combines the signals across the different arrays.

This describes a holistic system, that localises, separates and enhances the different sources in a room with ad-hoc distributed microphone arrays. Yet, for the system to be truly holistic, a couple of extra considerations should be taken care of. Firstly, all systems should be robust against (small) sample rate offers. In none of the systems described above combine signals from different arrays directly. The cluster informed separation also handles the (virtual) microphone inputs independently at first, which although not tested, could lead to robustness to sample rate offset. Secondly, the bandwidth is limited in WASNs, necessitating a lossy data transmission. However, [28] has shown that deep speech enhancement is able to reverse part of the distortions introduced by the codec. Lastly, the systems should be real time capable and run on edge. Therefore, causal and low latency DNN should be designed, with low computational complexity. Of course, this always comes at the cost of some performance loss. Future research should indicate to what extent the performance is kept.

The field of WASNs, and ad-hoc distributed microphone arrays is quite exciting. Many undiscovered algorithms are waiting to be uncovered, and many applications are waiting for their breakthrough. Going to a meeting room with your laptops and smart glasses and connecting them with each other to provide the best possible experience for a visual attendee. Or linking up all the microphones available in a cocktail party – phones for instance – to enhance the audio of hearing aid users so they can better enjoy these parties. As a matter of fact, there is no reason to stop only with hearing aids, but additionally enhance the audio of any hearable user in at the party, who wants to hear their friends more easily.

References

[1] A. Bohlender, A. Spriet, W. Tirry, and N. Madhu. *Neural networks using full-band and subband spatial features for mask based source separation*. In 2021 29th European Signal Processing Conference (EUSIPCO), pages 346–350. IEEE, 2021.

[2] A. Bohlender, A. Spriet, W. Tirry, and N. Madhu. *Spatially Selective Speaker Separation Using a DNN With a Location Dependent Feature Extraction*. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2023.

[3] T. Yoshioka, X. Wang, D. Wang, M. Tang, Z. Zhu, Z. Chen, and N. Kanda. *VarArray: Array-geometry-agnostic continuous speech separation*. In ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6027–6031. IEEE, 2022.

[4] K. Tesch and T. Gerkmann. *Spatially selective deep non-linear filters for speaker extraction*. In ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1–5. IEEE, 2023.

[5] K. Tesch and T. Gerkmann. *Multi-channel speech separation using spatially selective deep non-linear filters*. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 32:542–553, 2023.

[6] A. Bohlender, A. Spriet, W. Tirry, and N. Madhu. *Weakly doa guided speaker separation with random look directions and iteratively refined target and interference priors*. In 2024 International Workshop on Acoustic Signal Enhancement (IWAENC), pages 80–84. IEEE, 2024.

[7] S. Kindt, A. Bohlender, and N. Madhu. *Improved separation of closely-spaced speakers by exploiting auxiliary direction of arrival information within a u-net architecture*. In 2022 18th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pages 1–8. IEEE, 2022.

[8] J. Van Damme, S. Kindt, S. Song, J. Maes, and N. Madhu. *Investigation on system bandwidth for dnn-based binaural sound localisation for hearing aids*. In 2024 International Workshop on Acoustic Signal Enhancement (IWAENC), pages 26–30. IEEE, 2024.

[9] A. Van Den Oord, O. Vinyals, et al. *Neural discrete representation learning*. Advances in neural information processing systems, 30, 2017.

[10] A. Bohlender, A. Spriet, W. Tirry, and N. Madhu. *Exploiting temporal context in CNN based multisource DOA estimation*. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29:1594–1608, 2021.

[11] S. Chakrabarty and E. A. P. Habets. *Multi-Speaker DOA Estimation Using Deep Convolutional Networks Trained With Noise Signals*. IEEE Journal of Selected Topics in Signal Processing, 13(1):8–21, 2019. doi:10.1109/JSTSP.2019.2901664.

[12] S. Kindt, A. Bohlender, and N. Madhu. *2d acoustic source localisation using decentralised deep neural networks on distributed microphone arrays*. In Speech Communication; 14th ITG Conference, pages 1–5. VDE, 2021.

[13] Y. Gong, S. Liu, and X.-L. Zhang. *End-to-end two-dimensional sound source localization with ad-hoc microphone arrays*. In 2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pages 1944–1949. IEEE, 2022.

[14] L. Feng, Y. Gong, Z. Liu, X.-L. Zhang, and X. Li. *Learning Multi-dimensional Speaker Localization: Axis Partitioning, Unbiased Label Distribution, and Data Augmentation*. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2024.

[15] U. Kowalk, S. Doclo, and J. Bitzer. *Geometry-aware DoA Estimation using a Deep Neural Network with mixed-data input features*. In ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1–5. IEEE, 2023.

[16] A. J. Muñoz-Montoro, P. Vera-Candeas, and M. G. Christensen. *A Coherence-based Clustering Method for Multichannel Speech Enhancement in Wireless Acoustic Sensor Networks*. In 2021 29th European Signal Processing Conference (EUSIPCO), pages 1130–1134. IEEE, 2021.

[17] S. Kindt, M. Meeldijk, and N. Madhu. *Ad Hoc Distributed Microphones Clustering: A Comparative Analysis on Using Coherence and Signal-Specific Features*. In Speech Communication; 15th ITG Conference, pages 11–15. VDE, 2023.

[18] J. Kim, S. Kindt, N. Madhu, and H.-G. Kang. *Enhanced Deep Speech Separation in Clustered Ad Hoc Distributed Microphone Environments*. In Proc. Interspeech 2024, pages 1–5, 2024.

[19] S. Gergen, R. Martin, and N. Madhu. *Source separation by fuzzy-membership value aware beamforming and masking in ad hoc arrays*. In Speech Communication; 13th ITG-Symposium, pages 1–5. VDE, 2018.

[20] S. Rickard and O. Yilmaz. *On the approximate W-disjoint orthogonality of speech*. In 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing, volume 1, pages I–529. IEEE, 2002.

[21] E. A. Habets, J. Benesty, S. Gannot, and I. Cohen. *The MVDR beamformer for speech enhancement*. In Speech Processing in Modern Communication: Challenges and Perspectives, pages 225–254. Springer, 2010.

[22] J. Heymann, L. Drude, and R. Haeb-Umbach. *Neural network based spectral mask estimation for acoustic beamforming*. In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 196–200. IEEE, 2016.

[23] M. Souden, J. Benesty, and S. Affes. *On optimal frequency-domain multichannel linear filtering for noise reduction*. IEEE Transactions on audio, speech, and language processing, 18(2):260–276, 2009.

[24] H. Wang, A. Pandey, and D. Wang. *A systematic study of DNN based speech enhancement in reverberant and reverberant-noisy environments*. Computer Speech & Language, 89:101677, 2025.

[25] L. Becker, S. Kindt, and R. Martin. *Fuzzy-clustering-supported Assignment of Smart-Speaker-based Microphone Arrays to Acoustic Sources in Reverberant Acoustic Environments*. In Speech Communication; 15th ITG Conference, pages 230–234. VDE, 2023.

[26] L. Becker, K. Naame, and R. Martin. *Source signal capture in acoustic sensor networks based on robust beamforming and source-related cluster estimation*. In 2024 International Workshop on Acoustic Signal Enhancement (IWAENC), pages 255–259. IEEE, 2024.

[27] T. Gburrek, J. Schmalenstroeer, and R. Haeb-Umbach. *Geometry calibration in wireless acoustic sensor networks utilizing DoA and distance information*. EURASIP Journal on Audio, Speech, and Music Processing, 2021(1):25, 2021.

[28] H. Zhao and N. Madhu. *Bitrate-Informed Coded Speech Enhancement Model*. In 2024 International Conference on Electronics, Information, and Communication (ICEIC), pages 1–4. IEEE, 2024.

A

List of Acronyms

AAM	Additive Angular Margin
AMA	Averaged Modulation Amplitude
ASN	Acoustic Sensor Network
ASR	Automatic Speech Recognition
BLE	Bluetooth Low Energy
BM-CLA	Broadband Mixing Co-operative Localisation Architectures
CLA	Co-operative Localisation Architectures
CMR	Cepstral Modulation Ratios
CMS	Cepstral Mean Subtraction
CNN	Convolutional Deep Neural Network
CRUSE	Convolutional Recurrent U-net architecture for Speech Enhancement
DECT	Digital Enhanced Cordless Telecommunications
DFT	Discrete Fourier Transform
DNN	Deep Neural Network
DNSMOS	Deep Noise Suppression Mean Opinion Score
DOA	Directions Of Arrival
DPRNN	Dual-Path Recurrent Neural Network
DPT	Dual-Path Transformer

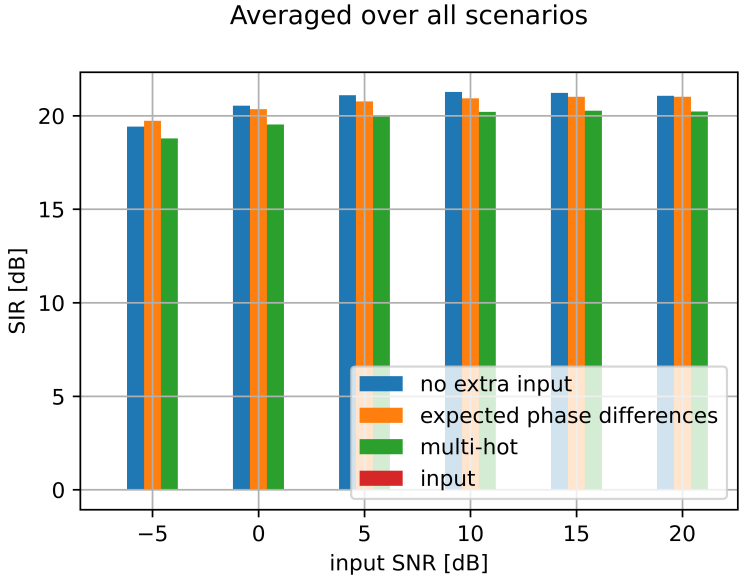
DPTNet	Dual-Path Transformer Network
DRINR	Direct-to-Reverberant-Interference-and-Noise Ratio
DRR	Direct-to-Reverberant Ratio
DSB	Delay and Sum Beamformer
DSP	Digital Signal Processing
DWS	Depth-Wise Separable
ECAPA-TDNN	Enhanced Propagation and Aggregation Time Delay Neural Network
eTAC	efficient Transform-Average-Concatenate
FC	Fully Connected
FCM	Fuzzy C-Means
FMV	Fuzzy Membership Values
FMVA-DSB	Fuzzy Membership Value Aware Delay and Sum Beamformer
GBF	Geometry-Based Features
GCC-PHAT	Generalized Cross-Correlation with Phase Transform
GRU	Gated Recurrent Unit
ICA	Independent Component Analyses
ILD	Interaural Level Differences
IoT	Internet of Things
ITD	Interaural Time Difference
IVA	Independent Vector Analyses
LBI	Location Based Input
LBO	Location Based Outputs
LC3	Low Complexity Communication Codec
LSTM	Long Short-Term Memory
MDCT	Modified Discrete Cosine Transform
MFCC	Mel Frequency Cepstral Coefficients
Mod-MFCC	Modulated Mel Frequency Cepstral Coefficients
MOS	Mean Opinion Score
MSC	Magnitude Squared Coherence
MSE	Mean Squared Error
MVDR	Minimum Variance Distortionless Response
MWF	Multichannel Wiener Filter
NM-CLA	Narrowband Mixing Co-operative Localisation Architectures

NMF	Non-negative Matrix Factorisation
PESQ	Perceptual Evaluation of Speech Quality
PIT	Permutation Invariant Training
POLQA	Perceptual Objective Listening Quality Assessment
PPM	Parts Per Million
PSD	Power Spectral Density
ReLU	Rectified Linear Unit
RIR	Room Impulse Response
RNN	Recurrent Neural Network
RT60	Reverberation Time
SBF	Signal-Based Features
SDLF	Source-Dependent Latent Features
SE	Squeeze-Excitation
SI-SDR	Scale Invariant Signal to Distortion Ratio
SIR	Signal-to-Interference Ratio
SMM	Spectral Magnitude Mask
SNR	Signal-to-Noise Ratio
SRO	Sample Rate Offsets
SRP-PHAT	Steered-Response Power with Phase Transform
STFT	Short-Term Fourier Transform
STO	Sample Time Offsets
STOI	Short-Time Objective Intelligibility
TAC	Transform-Average-Concatenate
TDAC	Time-Domain Aliasing Cancellation
TDOA	Time Difference Of Arrival
TF	Time-Frequency
VAD	Voice Activity Detection
VAE	Variational Auto-Encoder
VQ-VAE	Vector Quantised-Variational AutoEncode
WASN	Wireless Acoustic Sensor Network

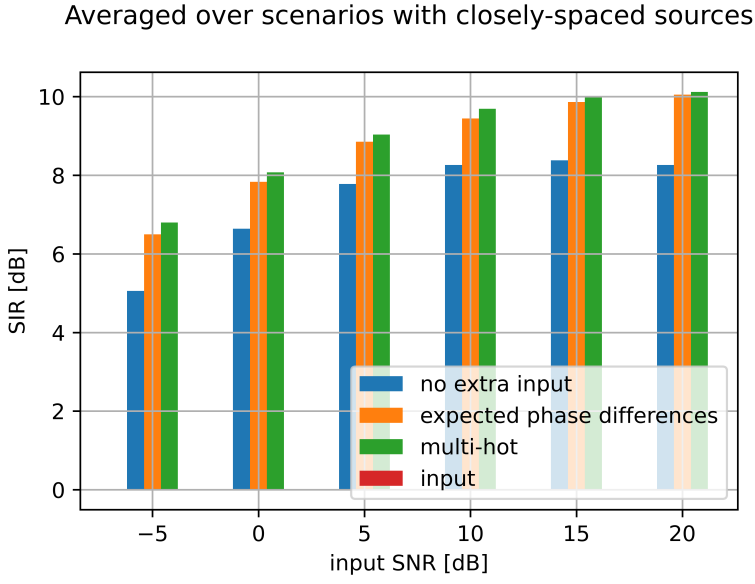
B

Absolute metrics for Chapter 3

This appendix adds some figures to the ones used in Chapter 3 with the goal to more easily compare the absolute values of the metrics with the other chapters of this book. Here, instead of using the difference (Δ) between the noisy and the improved versions for each method, the absolute values of the metrics of all the methods, as well as the input metrics. Note that the input SIR is zero, since the SIRs of both sources in the same scenario cancel out after averaging.

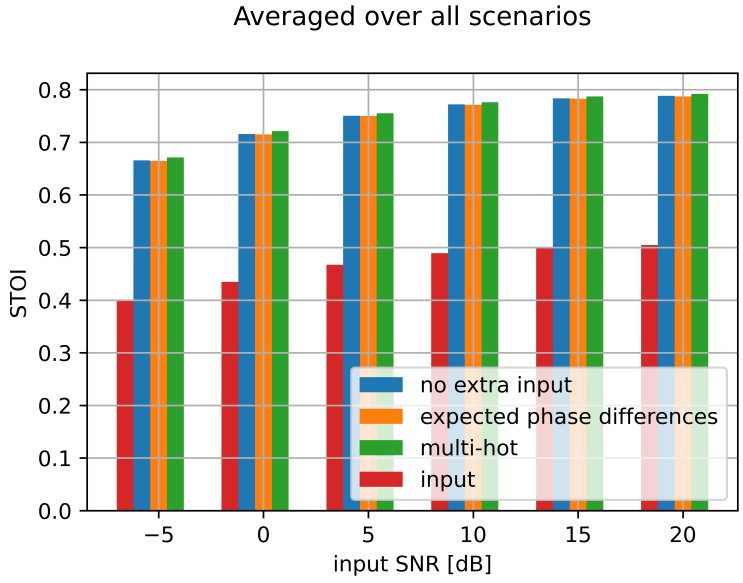


(a)

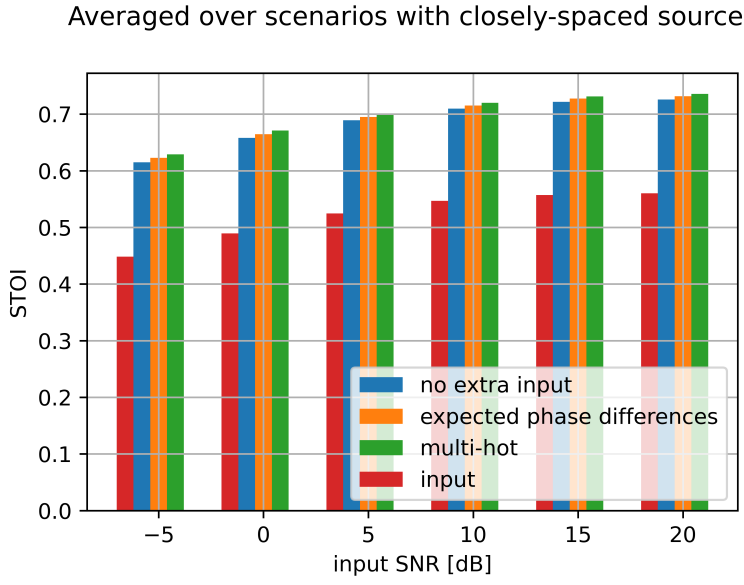


(b)

Figure B.1: The SIR metrics (as a function of different input SNRs) for all simulated cases on the top, and for the subset where sources are separated by only 20 degrees or less on the bottom. This is the counterpart to Figure 3.6a, where ΔSIR was plotted. Note that the input SIR is 0, since it is averaged over all speakers, where the two speakers of each scenario cancel each other out.

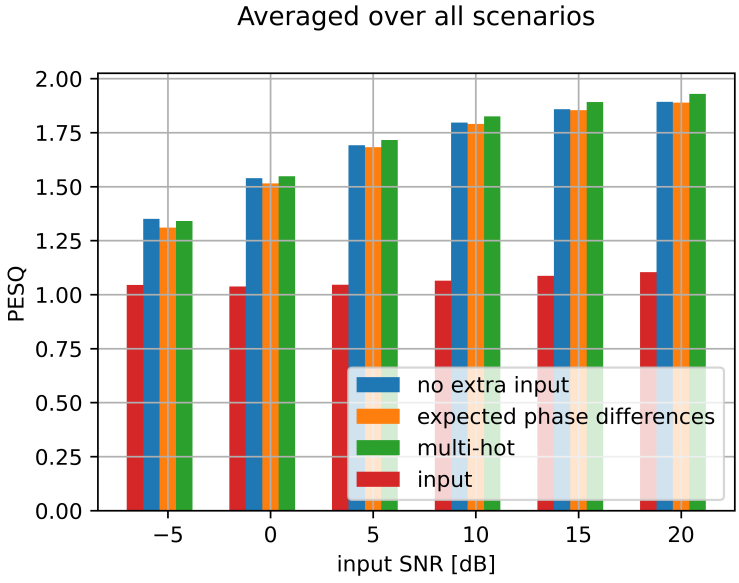


(a)

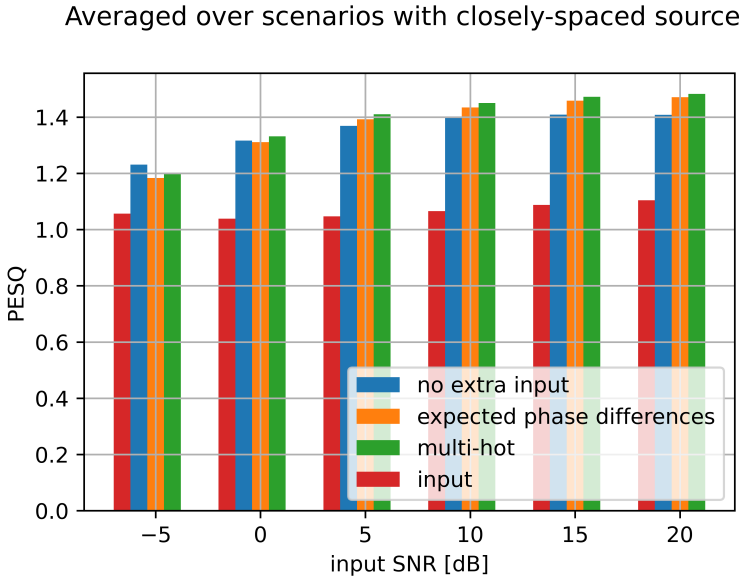


(b)

Figure B.2: The STOI metrics for all simulated cases (upper) and the subset with only closely spaced sources (lower). This is the counterpart to Figure 3.7a, where ΔSTOI was plotted.



(a)



(b)

Figure B.3: The PESQ metrics for all simulated cases (upper) and the subset with only closely spaced sources (lower). This is the counterpart to Figure 3.8a, where ΔPESQ was plotted.