AGH

AGH UNIVERSITY OF KRAKOW

FIELD OF SCIENCE ENGINEERING AND TECHNOLOGY

SCIENTIFIC DISCIPLINE AUTOMATION, ELECTRONICS, ELECTRICAL ENGINEERING AND SPACE TECHNOLOGIES

DOCTORAL THESIS

Towards discriminative speaker representations for speaker recognition and diarization

Author: Magdalena Marta Rybicka

Supervisors:
dr hab. inż. Konrad Kowalczyk
Associate Professor at AGH University of Krakow
dr Jesús Villalba
Assistant Research Professor at Johns Hopkins University

Completed in: AGH University of Krakow,
Faculty of Computer Science, Electronics and Telecommunications

Krakow, 2025

AGH

AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

DZIEDZINA NAUK INŻYNIERYJNO-TECHNICZNYCH

DYSCYPLINA AUTOMATYKA, ELEKTRONIKA, ELEKTROTECHNIKA
I TECHNOLOGIE KOSMICZNE

ROZPRAWA DOKTORSKA

Dyskryminatywne reprezentacje mówców dla zadań rozpoznawania mówców i diaryzacji

Autor: Magdalena Marta Rybicka

Promotorzy rozprawy:
dr hab. inż. Konrad Kowalczyk
Profesor uczelni Akademii Górniczo-Hutniczej im. Stanisława Staszica
dr Jesús Villalba
Assistant Research Professor na Johns Hopkins University

Praca wykonana: Akademia Górniczo-Hutnicza im. Stanisława Staszica
w Krakowie, Wydział Informatyki, Elektroniki i Telekomunikacji

Kraków, 2025

To Aunt Roma.

Abstract

The goal of speech processing systems is to handle natural, free-flowing speech across diverse conditions such as homes, offices, restaurants, and noisy cocktail parties. Each of these present unique challenges for system design. Building a universal and robust system capable of managing such a complex goal is achieved by combining different speech processing approaches, where each contributes with complementary types of information. A key trend towards the development of generic models is to integrate multiple tasks and leverage methods from other fields to extract sufficiently diverse information. Deep neural networks have significantly advanced speech technology systems, but research still continues to focus on improving performance, interpretability, and adapting cross-domain solutions.

This thesis investigates speaker representations for speaker recognition and diarization tasks, proposing optimization strategies to improve systems while retaining speaker-specific information. It further introduces explainable, discriminative representations for diarization and develops generic methods for joint diarization and separation, effective in both low- and high-overlap speech. The research contributions are presented through a series of five publications focused on speaker recognition and diarization tasks.

The first two of these publications focus on the speaker recognition task. The speaker recognition system involves several steps, the core of which is the extraction of speaker representations. In this thesis, the extraction method is based on a deep neural network approach. Two angular-based speaker objective functions are introduced, which adapt their hyperparameter values based on network performance and convergence in the current training step. Next, multiple proposals are presented for the architecture itself that increase processed temporal resolution, preserve and enhance processed information.

The next three publications concern the speaker diarization task. The development is based on an end-to-end approach to diarization, where the model directly estimates speaker activity from the input. An important procedure in this context is the estimation of the so-called attractors, which are representations of the speakers present in a given recording. In this thesis, the method of Non-Autoregressive Attractor (NAA) estimator is introduced. The approach estimates speaker representations for diarization by leveraging the properties of the embeddings present in the structure of the diarization model, providing a more explainable process of the attractor generation for the speaker diarization task, in contrast to the more obscure standard autoregressive method. The proposed NAA approach has been developed further and applied for joint speaker diarization and separation, at the same time aiming to bridge the gap between diarization and speech separation tasks.

Streszczenie

Celem systemów przetwarzania mowy jest możliwość przetwarzania naturalnej wypowiedzi w różnorodnych warunkach takich jak zacisze domowe, biura, restauracje, czy głośne imprezy. Każdy z zaprezentowanych scenariuszy stawia odmienne wyzwania podczas projektowania systemu. Opracowanie uniwersalnego i niezawodnego systemu zdolnego do realizacji tego złożonego zadania może zostać osiągnięte poprzez połączenie różnych systemów przetwarzania mowy, z których każdy wnosi komplementarne rodzaje informacji. Istotnym kierunkiem badań jest dążenie do systemów generycznych, poprzez łączenie wielu zadań i wykorzystywanie metod z innych dziedzin w celu uzyskania różnorodnych informacji. Głębokie sieci neuronowe (ang. deep neural networks) znacząco rozwinęły możliwości technologii mowy, niemniej jednak nadal trwają badania koncentrujące się na poprawie wydajności, interpretowalności oraz integracji rozwiązań z różnych dziedzin.

Niniejsza praca bada reprezentacje mówców dla zadań rozpoznawania mówców i diaryzacji, proponując strategie optymalizacji służące poprawie systemów przy jednoczesnym zachowaniu informacji charakterystycznych dla mówcy. W dalszej części wprowadza wyjaśnialne, dyskryminatywne reprezentacje dla diaryzacji i rozwija generyczne metody dla zadania jednoczesnej diaryzacji i separacji, skutecznych zarówno dla nagrań zawierających w niewielkim, jak i znaczącym stopniu mowę wielu osób wypowiadających się jednocześnie (ang. overlap speech). Wkład naukowy pracy jest zaprezentowany w postaci serii pięciu publikacji skupiających się na zadaniach rozpoznawania i diaryzacji mówców.

Pierwsze dwie publikacje dotyczą zadania rozpoznawania mówców. System rozpoznawania mówców zawiera kilka etapów, gdzie kluczowym jest ekstrakcja reprezentacji mówcy. W tej pracy metoda ekstrakcji opiera się na podejściu wykorzystującym głębokie sieci neuronowe. Zaproponowano dwie funkcje kosztu oparte na mierze kątowej, których wartości hiperparametrów adaptują się w zależności od poprawności odpowiedzi sieci oraz jej zbieżności w bieżącym kroku treningu. Następnie przedstawiono szereg propozycji dla samej architektury modelu, które pozwalają na zwiększenie przetwarzanej rozdzielczości czasowej oraz zachowują i wzmacniają przetwarzaną informację.

Kolejne trzy publikacje dotyczą zadania diaryzacji mówców. Rozwój metody jest oparty na podejściu typu end-to-end dla diaryzacji, gdzie model bezpośrednio estymuje aktywność mówców na podstawie informacji wejściowej. Ważną procedurą w tym zakresie jest estymacja tzw. atraktorów, czyli reprezentacji mówców występujących w danym nagraniu. W niniejszej pracy zaproponowano nieautoregresywną estymację atraktorów (ang. Non-Autoregressive Attractor (NAA) estimation). Podejście to wyznacza reprezentacje mówców poprzez wykorzystanie właściwości wektorów osadzeń (ang. embeddings) obecnych w architekturze modelu diaryzacji, zapewniając bardziej wyjaśnialny (ang. explainable) proces estymacji atraktorów dla zadania diaryzacji mówców, w przeciwieństwie do standardowego podejścia autoregresywnego. Opracowaną metodę rozszerzono i wykorzystano w zadaniu jednoczesnej diaryzacji i separacji mówców, aby zniwelować istniejącą lukę pomiędzy tymi procesami.

Streszczenie Rozszerzone

Wprowadzenie

W ostatnich latach możemy zaobserwować coraz więcej postępów technologicznych, które mają na celu ułatwienie i usprawnienie naszego życia, takich jak wirtualna i rozszerzona rzeczywistość (ang. virtual and augmented reality), Internet Rzeczy (ang. Internet of Things), „inteligentne” domy (ang. ‘smart’ homes) czy asystenci/agenci konwersacyjni wykorzystujący sztuczną inteligencję, np. ChatGPT, Google Gemini, Grok i wiele innych. Rozwiązania te są zaprojektowane w taki sposób, aby były łatwe i intuicyjne w użyciu. Jednym ze sposobów realizacji tego celu jest integracja technologii mowy, która może służyć jako naturalny środek do nawigacji i sterowania, a także stanowić istotne źródło informacji o użytkowniku lub prowadzonej konwersacji. W istocie, technologia mowy już teraz znajduje zastosowanie w wielu codziennych zadaniach, takich jak głosowe wybieranie numeru, dyktowanie i wypowiadanie komend dla inteligentnych asystentów, tłumaczeniu wypowiedzi na wybrany język w czasie rzeczywistym (ang. on-the-fly), częściowej automatyzacji obsługi klienta, czy też poprzez zapytania do asystentów głosowych (np. Siri lub Alexa). Zasadne jest również odniesienie się do niedawnej sytuacji pandemicznej, która wymusiła przejście znacznej części użytkowników na tryb pracy zdalnej i wykorzystywanie narzędzi audiowizualnych do prowadzenia spotkań. Narzędzia te implementują algorytmy detekcji aktywności głosowej oraz poprawy jakości sygnału mowy, co umożliwia uzyskanie lepszej jakości komunikacji. Aktualne kierunki badawcze koncentrują się na obszarze tzw. inteligencji konwersacyjnej (ang. conversational intelligence), w ramach której interakcje pojedynczego użytkownika z systemem przetwarzającym informacje w sposób pasywny ewoluują w stronę systemów aktywnie zaangażowanych zdolnych do realizacji złożonych zadań, takich jak automatyczne generowanie dokumentacji ze spotkań, weryfikacja faktów, ekstrakcja informacji o uczestnikach spotkania czy też wspieranie procesów tzw. uczenia się we współpracy (ang. collaborative learning) np. w grupach rówieśniczych. Tego typu systemy składają się z komponentów wykorzystujące różnorodne modalności, między innymi systemy dialogowe (ang. dialogue systems) odpowiedzialne za podążanie za kontekstem wypowiedzi, przetwarzanie obrazu wideo (ang. video processing) i ekstrakcja informacji, mechanizmy rozumowania oparte na wiedzy zdroworozsądkowej (ang. common-sense reasoning), a także moduły umożliwiające analizę konwersacji wieloosobowych.

Celem systemów przetwarzających mowę jest zdolność do analizy swobodnych wypowiedzi w różnorodnych warunkach, np. zacisza domowego, biura, restauracji, czy też przyjęcia (lub sytuacjach typu ‘cocktail party’), co jednocześnie stwarza szerokie spektrum wyzwań i uwarunkowań akustycznych. Aby opracować uniwersalny system przetwarzania mowy, który jest odporny na zróżnicowane warunki oraz zdolny do rozwiązywania złożonych zadań, konieczne jest połączenie różnych systemów zaprojektowanych do pojedynczych problemów. Na przykład, systemy do transkrypcji (ang. transcription systems) mogą składać się z połączenia systemów: poprawy jakości mowy (ang. speech enhancement) lub separacji mowy (ang. speech separation), automatycznego rozpoznawania mowy (ang. automatic speech recognition), rozpoznawania mówców (ang. speaker recognition) oraz diaryzacji mówców (ang. speaker diarization). Celem pierwszego z tych systemów, tj. poprawy jakości mowy, jest redukcja lub całkowite usunięcie niepożądanych inferencji, takich jak szum czy pogłos, które utrudniają zrozumienie i pogarszają jakość sygnału mowy. Zadaniem separacji mowy jest wydzielenie wypowiedzi jednej lub wielu osób do odrębnych strumieni audio. Automatyczne rozpoznawanie mowy służy do rozpoznawania wypowiedzianych słów (czyli ich transkrypcji). Rozpoznawanie mówców służy rozpoznaniu tożsamości danego mówcy. W ramach tego zadania wyróżnia się dwa podtypy: identyfikację oraz weryfikację. Pierwszy z nich polega na ustaleniu tożsamości danej osoby, z kolei drugi ma na celu potwierdzenie lub odrzucenie, czy dana osoba jest tym, za kogo się podaje. Diaryzacja mówców jest zadaniem polegającym na segmentacji nagrania wypowiedzi wieloosobowych na fragmenty, ze wskazaniem kiedy mówi ta sama osoba. Diaryzacja jest często wykorzystywana jako ogniwo łączące różne zadania, np. transkrypcja wywiadu otrzymana z systemu rozpoznawania mowy może dostarczać wypowiedzi mówców w postaci jednego ciągłego tekstu, bez rozróżnienia wypowiedzi poszczególnych mówców. W połączeniu z systemem diaryzacji tekst może zostać przyporządkowany do poszczególnych rozmówców – np. do osoby prowadzącej wywiad i respondenta – co pozwala na uzyskanie pełniejszej informacji i lepszego zrozumienia nagrania. W związku z tym, że niniejsza praca koncentruje się na reprezentacjach mówców w kontekście wybranych zadań systemów przetwarzania mowy, istotne jest podkreślenie znaczenia zadania rozpoznawania mówców, którego rozwiązania są często implementowane jako nowe metody dla innych obszarów, takich jak diaryzacja mówców, rozpoznawanie języka, rozpoznawanie emocji, czy też detekcja chorób na podstawie mowy.

Problemy oraz cele badawcze

Rozwiązania oparte na głębokich sieciach neuronowych (ang. deep neural networks) przyniosły znaczący przełom w możliwościach współczesnych systemów technologicznych. Niemniej jednak społeczność naukowa nadal podejmuje intensywne wysiłki w celu poprawy wydajności, zrozumienia oraz interpretacji decyzji podejmowanych przez głębokie sieci neuronowe. Jedną z popularnych praktyk jest przenoszenie rozwiązań z jednej dziedziny do innej (np. stosowanie metod przetwarzania obrazów w systemach przetwarzania mowy), co przyczynia się do postępu w nowym obszarze. Jednakże zdarza się, że proponowane metody nie są dostosowane do specyfiki domeny mowy. Kolejnym istotnym trendem w tej dziedzinie jest dążenie do budowy uniwersalnych systemów, które integrują wiele zadań, umożliwiając opracowanie generycznych modeli zdolnych do ekstrakcji różnorodnych informacji.

Niniejsza praca doktorska bada reprezentacje mówców w kontekście zadań rozpoznawania oraz diaryzacji mówców. Prezentuje ona metody zaprojektowane do poprawy systemów rozpoznawania mówców, ze szczególnym uwzględnieniem optymalnego procesu uczenia oraz zachowania cech i informacji charakterystycznych dla poszczególnych mówców. Praca dodatkowo wprowadza wyjaśnialne (ang. explainable) i dyskryminatywne reprezentacje mówców w kontekście zadania diaryzacji. Na zakończenie, w oparciu o poprzednie wyniki i wnioski, zaprezentowane są badania, których celem jest zapełnienie luki w domenie jednoczesnej diaryzacji oraz separacji mówców (ang. joint speaker diarization and separation) poprzez zaproponowanie generycznych metod, które działają skutecznie w warunkach naturalnej konwersacji, kiedy wypowiedzi nakładają się w sposób nieznaczny (ang. low-overlap speech) oraz w warunkach, kiedy wiele osób mówi niemalże jednocześnie (ang. high-overlap speech).

Opisane metody są ujęte w postaci następujących celów badawczych i hipotez:

1. Odpowiednie wartości hiperparametrów funkcji celu opartej na mierze kątowej (ang. angular-based loss function) poprawiają wydajność oraz zbieżność (ang. convergance) procesu uczenia systemu rozpoznawania mówców;

2. Zastosowanie cech wieloskalowych (ang. multi-scale features), zwiększenie rozdzielczości czasowej oraz uwzględnienie zależności częstotliwościowych przyczyniają się do poprawy skuteczności modeli rozpoznawania mówców;

3. Informacje zakodowane w wektorach osadzeń kodera na poziomie ramek (ang. frame-level encoder embeddings) modelu diaryzacji niosą względne informacje o mówcach i umożliwiają ich rozróżnianie w obrębie jednego nagrania;

4. Informacje o mówcach, wyekstrahowane przy użyciu metod zaproponowanych dla diaryzacji, mogą być wykorzystane w modelu jednoczesnej diaryzacji i separacji mówców, co pozwala na poprawę działania dla obu zadań.

Osiągnięcia badawcze pracy

Osiągnięcia i propozycje ukierunkowane na rozwiązanie przedstawionych problemów zostały zaprezentowane w formie serii pięciu publikacji obejmujących zagadnienia związane z rozpoznawaniem oraz diaryzacją mówców. Zaproponowane rozwiązania potwierdzają postawione hipotezy i wnoszą istotny wkład w rozwój metod rozpoznawania oraz diaryzacji mówców poprzez optymalizację podejść opartych na głębokich sieciach neuronowych.

Pierwsze dwie publikacje skupiają się na zadaniu **rozpoznawania mówców**. System rozpoznawania mówców zawiera kilka etapów, gdzie kluczowym jest ekstrakcja reprezentacji mówcy. W przypadku zadania weryfikacji, reprezentacje te są w kolejnych krokach przetwarzane i porównywane między sobą, zwracając wynik ich podobieństwa, który pozwala stwierdzić, czy nagrania pochodzą od tego samego mówcy, czy też nie. Obecnie ekstraktory reprezentacji mówców są oparte na głębokich sieciach neuronowych. W literaturze dużo uwagi jest poświęcone modyfikacjom i poprawie architektury sieci neuronowej, jej komponentów, uwzględniając również funkcję straty zastosowaną do treningu modelu. W niniejszej pracy zaproponowano dwie funkcje celu oparte o miarę kątową. Rodzina tych funkcji jest szeroko stosowana i wykazuje poprawę wydajności systemów rozpoznawania mówców poprzez wprowadzenie modyfikacji, które prowadzą do odpowiedniej separacji reprezentacji pochodzących od różnych mówców (ang. between-class distance) przy jednoczesnym dążeniu do jak najbardziej zbliżonych reprezentacji dla wypowiedzi pochodzących od tego samego mówcy (ang. within-class distance). Istotnym wkładem zaproponowanych funkcji kosztu jest automatyczna adaptacja hiperparameterów, których wartości dostosowują się do poprawności odpowiedzi sieci i jej zbieżności w bieżącym kroku treningu. Przedstawiona propozycja wpłynęła na poprawę skuteczności oraz przyspieszenie zbieżności treningu sieci dla zadania rozpoznawania mówców. Następnie zaproponowano wiele rozwiązań dla samej architektury sieci. W pierwszej publikacji zmodyfikowano powszechnie stosowaną strukturę, tzw. model rezydualny (ang. residual model), czyli ResNet [69], poprzez zwiększenie rozdzielczości czasowej oraz dostosowanie architektury do zadania rozpoznawania mówców, co skutkuje poprawą skuteczności. Druga z publikacji także wprowadza dalsze ulepszenia dla modelu rezydualnego. Ta powszechnie stosowana struktura może zostać scharakteryzowana jako model ze zmniejszającą się skalą (ang. scale-decreased design). Oznacza to, że informacja wraz z przetwarzaniem przez sieć jest jednostajnie podpróbkowywana. Tego typu przetwarzanie może się przyczynić do utraty części informacji. Aby temu zapobiec dla zadania mówców zaadoptowany został model ze skalą permutowaną (ang. scale-permuted design), tzw. SpineNet [42]. Struktura ze skalą permutowaną oznacza, że rozmiar przetwarzanej informacji, tzw. mapy cech (ang. feature map) może się dowolnie zwiększać lub zmniejszać w trakcie przetwarzania. Ponadto, zaimplementowana struktura pozwala na łączenie map cech wieloskalowych, na podstawie których generowana jest reprezentacja mówcy, podczas gdy w standardowym podejściu wykorzystywana jest ostatnia mapa cech, która jest najbardziej podpróbkowana. Przeprowadzono również badania, które pozwalają na integrację dodatkowych modułów w celu dalszej poprawy wydajności informacji uzyskanej z nagrania, a mianowicie Res2Net [52] oraz Time-Squeeze-and-Excitation (T-SE) [76, 87]. Oba moduły zostały pierwotnie zaproponowane dla zadania przetwarzania obrazów dla modeli o architekturze rezydualnej. Blok Res2Net modyfikuje bazowe bloki struktury rezydualnej w celu poszerzenia pola recepcyjnego (ang. receptive field) oraz umożliwia uchwycenie cech wieloskalowych o drobniejszej rozdzielczości w ramach pojedynczego bloku rezydualnego. Z kolei celem bloku T-SE jest rekalibracja zależności pomiędzy mapami cech wzdłuż wymiaru kanałów oraz częstotliwości.

Kolejne trzy publikacje dotyczą zadania **diaryzacji mówców**. Zaproponowane metody są oparte na podejściu diaryzacyjnym typu end-to-end, co oznacza, że model bezpośrednio określa wynik diaryzacji (aktywności poszczególnych mówców), na podstawie otrzymanych danych wejściowych. Kluczową cechą tego podejścia jest transformacja sekwencji wejściowej sieci na sekwencję wektorów osadzeń (ang. embedding sequence), gdzie na każdy zestaw cech przypada jeden wektor. W literaturze wektory osadzeń na poziomie ramek (ang. frame-level embeddings) były wykorzystywane na wiele sposobów w celu ekstrakcji informacji diaryzacyjnej, informacji na temat mówców, czy też do określania liczby mówców w nagraniu. W tym zakresie ważną procedurą jest estymacja atraktorów (ang. attractors), które określają reprezentacje mówców występujących w danym nagraniu. Najbardziej popularne jest podejście autoregresywne (ang. autoregressive), znane jako Encoder-Decoder-based Attractors [73], gdzie osadzenia na poziomie ramek są wykorzystane do oszacowania atraktorów. W niniejszej pracy zaproponowana została metoda nieautoregresywnej estymacji atraktorów (ang. Non-Autoregressive Attractor estimation - NAA). Główna różnica między podejściem autoregresywnym a nieautoregresywnym polega na tym, że w podejściu autoregresywnym kolejne atraktory estymowane są sekwencyjnie na podstawie poprzednich wyników, natomiast w podejściu nieautoregresywnym wszystkie atraktory wyznaczane są równocześnie. Zaprezentowane podejście wyznacza reprezentacje mówców dla zadania diaryzacji poprzez wykorzystanie właściwości osadzeń na poziomie ramek występujących w strukturze modelu diaryzacji, zapewniając bardziej wyjaśnialny proces estymacji atraktorów, w przeciwieństwie do standardowego podejścia autoregresywnego. Ważne, aby wspomnieć, że w literaturze zostały zaproponowane inne nieautoregresywne rozwiązania, jednakże po raz pierwszy w kontekście diaryzacji zostało ono zaproponowane przez autorkę tej pracy. Początkowo, metoda NAA była zaprezentowana jedynie na nagraniach zawierających dwóch mówców oraz z założeniem, że liczba mówców w nagraniu jest znana. W toku dalszych badań, procedura estymacji atraktorów została rozwinięta i rozszerzona poprzez zaproponowanie wariacji NAA, które pozwoliły na jej zastosowanie do warunków bardziej generycznych, obejmujących zmienną, większą niż dwa i nieznaną liczbę mówców. Ewaluacje systemów uwzględniały wiele baz danych, zarówno nagrań symulowanych, jak i rzeczywistych. Kolejnym krokiem w rozwoju metody NAA było jej zintegrowanie z modelem dla zadania **jednoczesnej diaryzacji i separacji mówców**. Badania miały na celu zniwelowanie luki pomiędzy zadaniami diaryzacji i separacji, które są w zasadzie bardzo podobne. Celem diaryzacji jest wskazanie aktywności każdego z mówców, co może być zaprezentowane poprzez wskazanie prawdopodobieństwa występowania danego mówcy w danej ramce czasowej. Z kolei separacja mówców, która polega na rozdzieleniu sygnałów audio odpowiadających poszczególnym mówcom, typowo jest wykonywana poprzez estymację masek, które mogą być interpretowane jako aktywności mówców w domenie czasowo-częstotliwościowej. W związku z tym zaproponowano rozwiązanie w formie pojedynczej struktury trenowanej jednocześnie dla obu zadań, które pozwala na ekstrakcję zarówno wyniku diaryzacji, jak i separacji. Metoda NAA została również z sukcesem zaimplementowana do tego zadania oraz rozszerzona dla nagrań o małym udziale mówców wypowiadających się równocześnie, reprezentujących spokojną konwersację (typowe dla zadania diaryzacji) oraz nagrań zawierających duży udział mowy osób wypowiadających się jednocześnie (typowe dla separacji), co udowadnia wszechstronność metody NAA.

Acknowledgements

During my journey towards a Ph.D. degree, I had the honour to be surrounded by many people that, in a direct and indirect way, shaped my growth during that time.

In the first place, I would like to express my sincerest gratitude to my supervisors: prof. Konrad Kowalczyk and prof. Jesús Villalba. I would like to thank prof. Konrad Kowalczyk for his strong mentorship throughout my Ph.D., whose optimism and faith in my abilities were pushing me to achieve more than I would believe myself, and who, with endless patience, helped and guided me through challenges both big and small. I am grateful to prof. Jesús Villalba for being an endless source of ideas, for generously sharing his broad knowledge, and for our consistent and fruitful collaboration over the years that contributed significantly to the development of this doctoral thesis.

I want to thank all my colleagues from AGH Signal Processing Group for how impactful they were at my development as a teacher and researcher, for creating such a supportive atmosphere, and making each of our lab integrations an immeasurably enjoyable time. I would like to give special thanks to dr Marcin Witkowski for the many conversations and valuable advice and guidance from the very beginning of my research journey. I would like to thank dr Jakub Gałka for introducing me to the world of speech processing technologies.

I also deeply acknowledge prof. Najim Dehak's group at CLSP JHU, with whom I spent a significant part of my Ph.D. I am especially grateful to prof. Najim Dehak, whom I have the honour to call my (unofficial) advisor, and who always treated me as one of his own Ph.D. students. I would also like to thank the other research professors Thomas Thebaud and Laureano Moro-Velázquez who each helped make our research discussions enlightening and joyful. I am deeply grateful to Piotr Żelasko who made this collaboration possible and through which I had the opportunity to meet so many inspiring people. I sincerely appreciate the collaborations with other Ph.D. students in the group, whose knowledge and insights have continually inspired me throughout my journey. I would especially like to thank Saurabhchand, Anna and Sonal for our lab chats.

Beyond my academic environment, there are several people in my life that I feel truly blessed to have the support of. Firstly, I would like to thank my parents, Marta and Jan, who put tremendous effort into my education and have been there for me through all the ups and downs. I am incredibly grateful to my brother Maciek and sister-in-law Natalia, who, despite sometimes being thousands of kilometres away, were always present with love, understanding, and advice.

I am thankful to my friends, Julitta and Filip, with whom I've shared countless adventurous trips, lazy weekends, and much mirth. And to my friends Magda, Janek, Maneesha, Kelsey, Ania, Sylwia for their sincere friendship and for reminding me of what truly matters. I am grateful to all the friends with whom I shared and played music; those moments brought me joy, balance, and inspiration.

Last, but definitely not least I would like to thank Kyle, my best friend and love. I am deeply grateful for his continual encouragement to grow and for inspiring me to courageously and confidently pursue my dreams and goals.

Table of contents

Nomenclature **xxi**

1 Introduction **1**

1.1 Motivation and relevance of the research 1

1.2 Problem statement and research objectives 3

1.3 Contributions 5

1.4 Author's research in the topic 6

1.5 Outline of the Thesis 13

2 Research background **15**

2.1 Speaker recognition 15

2.1.1 Speaker verification system processing steps 17

2.1.2 Deep Neural Network-based speaker modeling 22

2.2 Speaker diarization 30

2.2.1 Cluster-based diarization 31

2.2.2 End-to-end speaker diarization 34

2.2.3 Speaker diarization and separation 38

2.3 Evaluation metrics 41

2.3.1 Speaker verification 41

2.3.2 Speaker diarization 44

2.3.3 Speaker separation 45

2.4 Datasets 45

2.4.1 Speaker recognition 45

2.4.2 Speaker diarization 47

2.4.3 Speaker diarization and separation 49

3 Overview of conducted research and main contributions **51**

3.1 Discriminative speaker representations 51

3.2 Non-Autoregressive Attractor estimation 60
3.2.1 Speaker diarization 60
3.2.2 Joint speaker diarization and separation 67

4 Finale 71
4.1 Summary 71
4.2 Future work 72

Bibliography 75

Appendix: Publication Series - Full Texts 91
I On Parameter Adaptation in Softmax-Based Cross-Entropy Loss for Improved Convergence Speed and Accuracy in DNN-Based Speaker Recognition 92
II Spine2Net: SpineNet with Res2Net and Time-Squeeze-and-Excitation Blocks for Speaker Recognition 98
III End-to-End Neural Speaker Diarization with an Iterative Refinement of Non-Autoregressive Attention-based Attractors 104
IV End-to-End Neural Speaker Diarization with Non-Autoregressive Attractors 110
V Joint Diarization and Separation Using SepFormer with Non-Autoregressive Attractors 126

Nomenclature

Acronyms / Abbreviations

AAS	Additive Angular Margin Softmax
AdaCos	Adaptively Scaling Cosine Logits
AHC	Agglomerative Hierarchical Clustering
AMS	Additive Margin Softmax
AS	Angular Softmax
ASR	Automatic Speech Recognition
BLSTM	Bidirectional Long Short-Term Memory
CDS	Cosine Distance Scoring
CSS	Continuous Speech Separation
DCT	Discrete Cosine Transform
DER	Diarization Error Rate
DET	Detection Error Tradeoff
DNN	Deep Neural Network
DPRNN	Dual-Path Recurrent Neural Network
EDA	Encoder-Decoder-based Attractor
EEND-EDA	End-to-End Neural Diarization with Encoder-Decoder Based Attractors
EEND-GLA	End-to-End Neural Diarization with Global and Local Attractors
EEND-NAA	End-to-End Neural Diarization with Non-Autoregressive Attractors
EEND-SS	Joint End-to-End Neural Speaker Diarization and Separation

EEND-VC	End-to-End Neural Diarization with Vector Clustering
EEND	End-to-End Neural network-based speaker Diarization
EER	Equal Error Rate
ETDNN	Extended Time Delay Neural Network
FFT	Fast Fourier Transform
FPR	False Positive Ratio
FRR	False Rejection Ratio
FTDNN	Factorized Time Delay Neural Network
GMM	Gaussian Mixture Model
JER	Jaccard Error Rate
JFA	Joint Factor Analysis
LDA	Linear Discriminant Analysis
LDE	Learnable Dictionary Encoding
LSTM	Long Short-Term Memory
MFCC	Mel-Frequency Cepstral Coefficients
minDCF	Minimal Detection Cost Function
mR18	modified ResNet-18
NAA	Non-Autoregressive Attractor
NAP	Nuisance Attribute Projection
NIST	National Institute for Standards in Technology
PIT	Permutation Invariant Training
PLDA	Probabilistic Linear Discriminant Analysis
SAD	Speech Activity Detection
SC-EEND	Speaker-wise Conditional End-to-End Neural Diarization
SCD	Speaker Change Detection
SE	Squeeze-and-Excitation

SI-SDRi	Scale-Invariant Signal-to-Distortion Ratio improvement
SID	Speaker Identification
SRE	Speaker Recognition Evaluation
SR	Speaker Recognition
SSAD	Single Speaker Activity Detection
SSGD	Speech Separation Guided Diarization
SSND	Speaker Separation via Neural Diarization
SVM	Support Vector Machines
SV	Speaker Verification
T-SE	Time-Squeeze-and-Excitation
TDNN	Time Delay Neural Network
TS-SEP	Target-Speaker Separation
TS-VAD	Target-Speaker Voice Activity Detection
UBM	Universal Background Model
VAD	Voice Activity Detection
WCCN	Within Class Covariance Normalization

Chapter 1

Introduction

1.1 Motivation and relevance of the research

Technological advancements are driven by the need to make our lives better. The rapid development of modern hi-tech solutions has led to the appearance of technologies such as virtual reality, Internet of Things, ‘smart’ homes and many more. Many proposed advances are developed so that they are interactive and natural to use. Speech is the most natural way of communication between humans, thus, it often constitutes the crucial part of advanced solutions. Since the development of the first speech recognition systems over 50 years ago, speech technology innovations and solutions have been integrated into our daily lives via applications such as voice dialing, translating spoken expressions on-the-fly, semi-automated customer service, or asking Siri or Alexa for a weather forecast. Let us also take into consideration the recent pandemic situation, which forced many people to work remotely and to use audio-video tools for on-line meetings. A lot of us are unaware that these tools have active voice activity detection and speech enhancement algorithms to facilitate meetings at a high quality (e.g. Cisco Webex, Microsoft Teams). Currently, the research direction is focused on conversational intelligence. Single-user interactions with passive systems evolve into actively engaging systems that enable completing complex tasks such as generating documentation during meetings (including medical records at doctor appointments), verifying facts, extracting information about meeting participants, or supporting collaborative learning within peer groups. Such systems are composed of intelligent components that span over diverse modalities, i.a. dialogue systems for context handling, video processing for information extraction, common-sense reasoning, analysis and recognition for multi-speaker conversations.

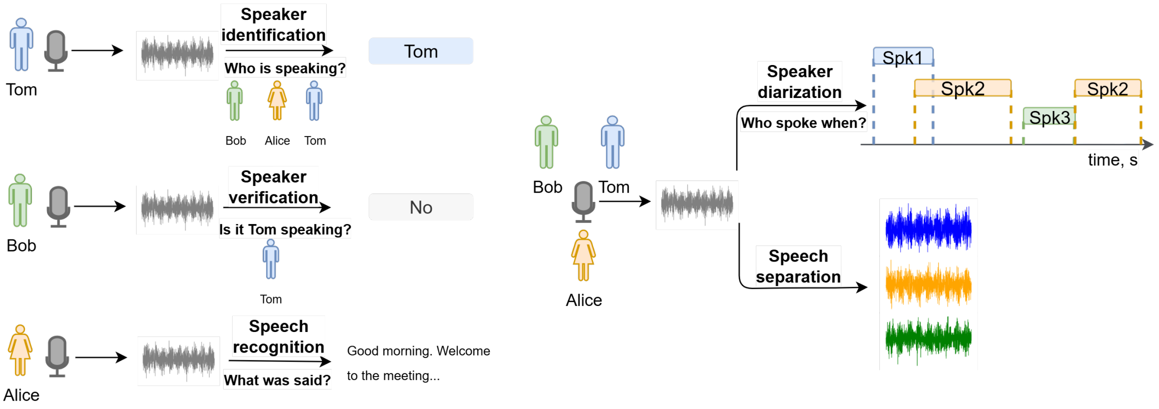


Figure 1.1: The general depiction of examples of the selected speech processing tasks.

One of the common goals and challenges of speech processing systems is to effectively handle free-flowing speech across diverse conditions and scenarios, such as home environments, offices (e.g. business meetings), restaurants, and cocktail parties. These conditions in fact provide a huge variety of speech characteristics due to the amount and type of noise, the amount of speech, recording length, the possibility of the occurrence of overlapping speech, multiple number of speakers, background speech, the type of device used, or even the character of the conversation (which influences the amount of silence, speaker turns, speech overlap, and the number of speakers). Obtaining high-quality and accurate results often involves an ensemble of multiple systems. For example, in transcription pipelines this may include automatic speech recognition (ASR), speaker recognition (SR), and speaker diarization. Furthermore, since overlapping speech presents a significant challenge for these systems, speaker separation or other similar enhancement techniques are often used to improve performance.

The mentioned speech processing systems represent different tasks, each addressing a specific aspect of analysing and understanding spoken language. Their general scheme is presented in Figure 1.1. ASR is a task that answers the question "what was said". Speaker recognition, which is represented by subtasks of speaker identification and verification, answers "who is speaking" (identification) or "are the words spoken by the particular person" (verification). Diarization is the task of segmenting audio into speaker-specific segments, answering the question "who spoke when". Diarization model often facilitates the combination of speech and speaker recognition. The goal of speaker separation is to isolate and extract the speech of multiple speakers into separate individual audio streams. It should be noted that such systems often constitute a preprocessing step for spoken language understanding systems whose goal is to extract the meaning of the spoken sentence, and represent another human-computer interaction technology.

One of the key aims of current speech technology development is to process conversations, particularly those occurring ’in the wild’ with multiple speakers involved. Such multi-speaker interactions fall under the so-called cocktail-party problem, which includes not only conversational speech but also situations with heavy speech overlap and background noise. Deep neural network (DNN) based representations have brought significant improvement and possibilities for speech applications. Nevertheless, their reliability can still be improved through further research. The most prominent challenges include adverse acoustic conditions in the recorded signal e.g. room reverberation, environmental noise and background speech, overlapping speech, different domains among recordings (e.g. the microphone and telephone speech often have different sampling frequency), short duration of recordings, and others. The described difficulties have been addressed in different conference challenges, which emphasize the importance of the presented problems, e.g. Short-duration Speaker Verification Challenge [204], Far-Field Speaker Verification Challenge [148], VoxSRC Challenge [121] - speaker verification and diarization, VOiCES from a Distance Challenge [124], CHiME Challenge [193] - speech separation, recognition and diarization in multi-speaker scenario, DIHARD Speech Diarization Challenge [161], DISPLACE Challenge [86] - speaker diarization, language diarization, speech recognition on multilingual data.

The main theme of the thesis is centred around speaker representations, investigated from the perspectives of speaker recognition and diarization tasks. For this reason, it is important to emphasize that the advances in the field of speaker recognition are versatile and often contribute to the milestones in other speech and signal processing domains. Let us consider as evidence the example of x-vectors [175] - one of the state-of-the-art speaker embeddings (representations) - which soon after its proposal has been successfully applied for speaker diarization [167], language recognition [172], and emotion recognition [164] or even for estimation of room acoustics [92]. Several recent works employ x-vectors for health applications, e.g. Parkinson’s disease detection [83, 120] or Alzheimer’s detection [139, 143] from speech.

1.2 Problem statement and research objectives

In recent decades, deep neural networks (DNNs) have provided breakthroughs in many fields of speech processing technology, due to access to large amounts of data, the availability of better computational resources, as well as due to the contribution of the research community and proposals of methods that further develop existing techniques. However, proposed methods often are not properly adjusted for the particular speech task as they are typically borrowed from other signal processing domains. At the same time, as the systems become better tailored for single tasks, the research community tries to address and develop more generic systems that handle multi-modal information and leverage mutual properties to achieve better performance, robustness to adverse conditions, and capability to interpret the real-life conditions environment at least on par with human accuracy.

The research presented in this thesis explores various aspects of speaker characterization systems, with a primary focus on investigating speaker representations across different tasks. It presents optimization methods that are tailored for better speaker recognition systems, focused on optimal training and preservation of speaker information. Next, it introduces explainable and discriminative speaker representations for the speaker diarization task. Finally, using the proposals of the previous work, this thesis tries to address the joint speaker diarization and separation task, by proposing generic methods that are efficient in both conversational (low speech overlap) and highly overlapped speech conditions, for both simulated and real-life recordings.

Described problems lead to the following research objectives and hypotheses, focusing on several aspects of speaker characterization systems:

1. Proper hyperparameter values of the angular-based objective function improve the performance and training convergence of speaker recognition system;

2. Multi-scale features, increasing the temporal resolution, and focusing on frequency dependencies improves the effectiveness of speaker recognition;

3. The frame-level encoder embeddings carries relative speaker information and enables discrimination between them within a recording;

4. Relative speaker information extracted with methods proposed for diarization can be successfully employed in joint speaker diarization and separation enabling improved performance for both tasks.

1.3 Contributions

The research contributions present advancements and optimizations of the structures for speaker recognition, speaker diarization, and joint speaker diarization and separation tasks. This thesis presents its contributions in the form of a publication series, discussed in further sections of this document:

I. **M. Rybicka** and K. Kowalczyk, "*On Parameter Adaptation in Softmax-Based Cross-Entropy Loss for Improved Convergence Speed and Accuracy in DNN-Based Speaker Recognition*", Interspeech, Shanghai, China, 2020.

II. **M. Rybicka**, J. Villalba, P. Żelasko, N. Dehak, K. Kowalczyk, "*Spine2Net: SpineNet with Res2Net and Time-Squeeze-and-Excitation Blocks for Speaker Recognition*", Interspeech, Brno, Czech Republic, 2021.

III. **M. Rybicka**, J. Villalba, N. Dehak and K. Kowalczyk, "*End-to-End Neural Speaker Diarization with an Iterative Refinement of Non-Autoregressive Attention-based Attractors*", Interspeech, Incheon, South Korea, 2022.

IV. **M. Rybicka**, J. Villalba, T. Thebaud, N. Dehak and K. Kowalczyk, "*End-to-End Neural Speaker Diarization with Non-Autoregressive Attractors*", IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2024.

V. **M. Rybicka**, K. Kowalczyk, T. Thebaud, N. Dehak, J. Villalba, "*Joint Diarization and Separation Using SepFormer with Non-Autoregressive Attractors*", IEEE Signal Processing Letters, 2025.

In the following part of this thesis these publications will be referenced by their Roman numerals. The full texts are included in the Appendix of the thesis.

The main contributions are in the areas of speaker recognition, speaker diarization and joint speaker diarization and separation and can be summarized as follows:

- Proposal of an angular-based speaker recognition objective function with adaptive hyperparamters, which is a softmax-based cross-entropy loss function that adapts its hyperparameters based on neural network's performance and convergence at the current training step. The proposed approach improves performance and speeds up network convergence for the speaker recognition task;

- Adaptation and modification of ResNet architecture by increasing temporal resolution of the model for improved speaker recognition performance;

- Adaptation of the scale-permuted architecture to preserve speaker information, along with an exploration of modules that enhance model resolution for the speaker recognition task;
- Proposal of Non-Autoregressive Attractor estimator for the speaker diarization task – a method for estimating speaker representations for diarization, which allows to leverage properties of the frame-level embeddings present in the structure of the diarization model, providing a more explainable attractor generation in speaker diarization;
- Extension of Non-Autoregressive Attractor method to generic conditions of a flexible and unknown number of speakers, presented for several simulated and real scenarios for the speaker diarization task;
- Extension of the separation model into a unified framework for joint speaker diarization and separation, enabling efficient and effective performance of these two tasks;
- Incorporation of the Non-Autoregressive Attractor estimation method for joint speaker diarization and separation, with adaptation to both conversational and high-overlap speech conditions, demonstrating competitive or superior performance compared to models designed for either the joint task or the individual subtasks.

1.4 Author's research in the topic

In addition to the Publications presented in Section 1.3, the author has co-authored other papers related to the field, participated in challenges, and contributed to various research projects. This section provides an overview of the author's additional work beyond the main contributions of this thesis.

Conference Papers:

1. M. Witkowski, **M. Rybicka** and K. Kowalczyk, "*Speaker Recognition from Distance Using X-Vectors with Reverberation-Robust Features*", 2019 IEEE Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA), Poznan, Poland, 2019;

The paper presented research in the field of speaker recognition, focusing on reducing the negative impact of reverberation by exploring reverberation-robust features (acoustic representations of the audio).

2. M. Witkowski, **M. Rybicka** and K. Kowalczyk, "*Sparse Linear Prediction-based Dereverberation for Signal Enhancement in Distant Speaker Verification*", 2021 IEEE European Conference on Signal Processing (EUSIPCO), Dublin, Ireland, 2021;

   The paper introduced a novel dereverberation method of audio recording, which was investigated as a preprocessing step for robust speaker recognition. The method was examined with the state-of-the-art speaker modeling method, as well as the method proposed in the Publication I.

3. J. Villalba, B. J. Borgstrom, S. Kataria, **M. Rybicka**, C. D. Castillo, J. Cho, L. P. García-Perera, P. A. Torres-Carrasquillo, N. Dehak "*Advances in Cross-Lingual and Cross-Source Audio-Visual Speaker Recognition: The JHU-MIT System for NIST SRE21*", Odyssey, Beijing, China, 2022;

   The paper presented a description of the speaker recognition system for NIST SRE21 evaluation, developed by the team of researches from Johns Hopkins University (JHU) and Massachusetts Institute of Technology (MIT). Speaker Recognition Evaluation (SRE), organized by the National Institute of Standards and Technology (NIST), is one of the most important evaluations of speaker recognition systems. It attracts participants from universities and companies worldwide, aiming to advance and evaluate the state-of-the-art in speaker recognition. This evaluation addressed speaker recognition using multilingual conversational telephone speech (CTS) and audio from video (AfV). It included multimodal tracks with cross-source verification by comparing speaker identities across CTS and AfV recordings, as well as cross-language trials, where the system verified whether the recordings spoken in different languages belong to the same speaker. The author's contribution to this publication involved the investigation and development of invariant representation learning (IRL), aimed at making the speaker recognition system invariant and robust to language mismatch.

4. S. Kacprzak, **M Rybicka**, and K. Kowalczyk, "*Spoken Language Recognition with Cluster-Based Modeling*", IEEE 2022 International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 2022.

   The paper investigated the language recognition problem, which is a topic related to speaker recognition. The presented research used unsupervised cluster-based modeling in the language recognition system, which served as an approach to construct multiple independent language models.

Technical reports and challenges

1. VOiCES from a Distance Challenge 2019, Fixed Condition of Speaker Recognition task (2019)

   The VOiCES from a Distance Challenge 2019 aimed to advance research in speaker recognition and automatic speech recognition (ASR), with a particular focus on single-channel far-field audio in diverse noisy environments. The author and her team (researchers from AGH University of Krakow) participated in the Fixed Condition of the Speaker Recognition task, where a speaker recognition system is developed using only a predefined, limited (i.e. fixed) set of training data.

2. Short-duration Speaker Verification Challenge 2020, Task 2: Text-independent Speaker Verification (2020)

   This challenge focused mainly on speaker verification systems for short recordings. The author and her team (researchers from AGH University of Krakow) participated in Task 2: Text-independent Speaker Verification, which means that the speaker verification was conducted regardless of the words spoken in the compared recordings (which is contrary to the text-dependent speaker verification). In this task, the recordings for creating the speaker reference model ranged from 3 to 120 seconds in duration, while the test recordings ranged from 1 to 8 seconds. The task presented an additional challenge by including two scenarios: (1) both the model and test recordings were in the same language (Persian), and (2) the model and test recordings were in different languages (Persian and English). The submitted system included the speaker modeling architecture method proposed in Publication I.

3. NIST 2021 Speaker Recognition Evaluation (SRE21) (2021)

   The author participated in the challenge as a part of the research team from Johns Hopkins University, during the research visit in 2021. The team included researchers from Johns Hopkins University (JHU) and Massachusetts Institute of Technology (MIT) and resulted in the joint article, described above (subsection Conference Papers: *"Advances in Cross-Lingual and Cross-Source Audio-Visual Speaker Recognition: The JHU-MIT System for NIST SRE21"*). The challenge itself has also been summarized in the description provided in the referenced article.

4. Second Multimodal Information Based Speech Processing (MISP) Challenge, Track 1, Audio-Visual diarization systems (2022)

   The author participated in the challenge as a part of the research team from Johns Hopkins University, during the research visit in 2022. The challenge, part of the ICASSP 2022 Signal Processing Grand Challenge, focused on speech processing in home scenarios with multiple simultaneous conversations, background noise, and reverberation. It promotes the use of both audio (far-, middle-, and near-field) and video (far- and middle-field) modalities to improve performance under these challenging conditions. Track 1 specifically addressed Audio-Visual Speaker Diarization. The proposed system incorporated an approach introduced in Publication III.

Conference presentations

1. **09/2019**: poster presentation at 2019 IEEE Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA), Poznan, Poland.
   Paper presented: M. Witkowski, M. Rybicka and K. Kowalczyk, "*Speaker Recognition from Distance Using X-Vectors with Reverberation-Robust Features*".

2. **10/2020**: oral presentation at Interspeech 2020 conference, Shanghai, China (conference was held as virtual due to pandemic situation).
   Paper presented: M. Rybicka and K. Kowalczyk, "*On Parameter Adaptation in Softmax-Based Cross-Entropy Loss for Improved Convergence Speed and Accuracy in DNN-Based Speaker Recognition*".

3. **08/2021**: oral presentation at Interspeech 2021 conference, Brno, Czech Republic.
   Paper presented: M. Rybicka, J. Villalba, P. Żelasko, N. Dehak, K. Kowalczyk, "*Spine2Net: SpineNet with Res2Net and Time-Squeeze-and-Excitation Blocks for Speaker Recognition*".
   For this conference, the author was awarded with Travel Grant, which is presented to students and young scientists to support their participation in the conference. Award granted by the organizers of Interspeech 2021 and the International Speech Communication Association (ISCA).

4. **09/2022**: poster presentation at Interspeech 2022 conference, Incheon, South Korea.
   Paper presented: M. Rybicka, J. Villalba, N. Dehak and K. Kowalczyk, "*End-to-End Neural Speaker Diarization with an Iterative Refinement of Non-Autoregressive Attention-based Attractors*".

5. **04/2025**: poster presentation at 2025 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Hyderabad, India.
   Paper presented: M. Rybicka, J. Villalba, T. Thebaud, N. Dehak and K. Kowalczyk, "*End-to-End Neural Speaker Diarization with Non-Autoregressive Attractors*".

Projects

1. **2018-2021**: Audio Processing using Distributed Acoustic Sensors (APDAS), First TEAM Program, Foundation for Polish Science (FNP)

   This research project aimed to develop methods for distributed signal processing in order to enable acoustic scene analysis. The goal was the enhancement of speech intelligibility in hands-free communication and robust voice-based human-computer interfaces over a distance. The authors role in the project was algorithm implementation, evaluation, and result analysis in the field of deep neural network based speaker recognition, collaboration with other project members in order to develop robust speaker recognition system in reverberant conditions, and research in the area of speaker diarization.

2. **2022**: Machine Learning for Spatial Audio Processing (MLSAP), OPUS Program, National Science Centre (NCN)

   The project investigated how classical signal processing and machine learning could be integrated to process audio and speech more effectively. It aimed to design innovative methods that harnessed the complementary strengths of both approaches, with the goal of advancing sound event detection and localization, signal extraction, and the classification of speech and audio signals. The authors role was development of algorithms, their implementation, evaluation, and analysis of results in the field of speaker recognition and diarization based on deep neural networks.

3. **2024-2025**: Intelligent voice assistant for managing medical records according to doctor recommendations (LINGE), INFOSTRATEG IV Program, Narodowe Centrum Badań i Rozwoju

   The goal of the project is creating a universal system designed to support and simplify physicians' work by automating the completion of Electronic Medical Records and related documents, such as prescriptions, referrals, and sick leave forms. The system integrates voice-based modules for automatic speech recognition and natural language processing with the medical form solutions developed by the project partner. The expected result is less time spent on completing medical records, allowing doctors to focus more on in-depth patient consultations. The authors role in the project was algorithm implementation, evaluation, and result analysis in the field of diarization and target-speaker voice activity detection (speech activity detection of a specific person).

4. **2024-2025**: Acoustic Intelligence – towards self-supervised deep neural acoustic analysis (Acoustic Intelligence), OPUS Program, National Science Centre, Poland (NCN)

   The project explores the use of self-supervised learning for audio, allowing models to learn from unlabeled data and identify patterns with higher robustness than traditional supervised methods. Its goal is to develop innovative approaches, such as Universal Audio Representation, Universal Acoustic Analysis, and Universal Constituent Audio Signal Enhancement, to build intelligent systems capable of independent improvement across a wide range of audio tasks. Additionally, the project goal is to investigate the domains of self-supervised acoustic signal enhancement and acoustic scene analysis. The author's role was development of algorithms, their implementation, evaluation, and analysis of results in the field of speaker separation based on deep neural networks.

Research visits

During the Ph.D. pursuit, the author established collaboration and visited prof. Najim Dehak's group at Center for Language and Speech Processing (CLSP), Johns Hopkins University (JHU), Baltimore, USA, and worked under supervision of prof. Jesús Villalba and prof. Najim Dehak. The visits were organized as part of the projects:

1. **09/2021 - 11/2021** – Audio Processing using Distributed Acoustic Sensors project, funded by the Foundation for Polish Science within the First TEAM Program;

2. **10/2022 - 08/2023** – Fulbright Junior Research Award;

3. **11/2023 - 09/2024** – funding international research visits of young employees and doctoral students, Excellence Initiative - Research University program, AGH University of Krakow.

The collaboration was initiated in 2020 and since then has resulted in 3 research visits and 5 joint publications. During this period, the author was in continuous contact with professors, holding weekly meetings and consultations on research progress, either in person or remotely.

During my research visits, I had the opportunity to participate in regular team meetings, where I presented my work and became familiar with the research of other doctoral students. During the first visit, the research focused on the development of the speaker recognition task. I had the opportunity to be part of the JHU-MIT team at the NIST SRE21 evaluation and work on the robustness of the speaker recognition system to multilingual speaker recording representations.

The next 10-month visit was with a project awarded with Fulbright Junior Research Award. Fulbright is one of the largest exchange programs with the United States. The Fulbright Junior Research Award provides support and funding for Ph.D. students to perform a research project at scientific institutions in the USA. During this stay, I worked on developing the diarization system, initially proposed in the Interspeech 2022 conference paper (which was also the result of a collaborative effort) and began preparing a manuscript for the IEEE/ACM Transactions on Audio, Speech, and Language Processing. I also joined the JHU team in the Multimodal Information Based Speech Processing (MISP) Challenge 2022 (described in more detail above) as part of this research visit. In addition to my main research work, I had the opportunity to actively participate in scientific events organized at CLSP. I volunteered during the preparations for the 2023 Frederick Jelinek Memorial Summer Workshop on Speech and Language Technologies (JSALT), an annual multi-week workshop that produces numerous innovative solutions in the field of speech processing. At the end of the research visit, I had the opportunity to share my research findings and conclusions by giving a talk about my research at the Human Language Technology Center of Excellence, JHU.

My third stay, which was an 11-month extension of the project, focused on continuing work on diarization and initiating a new research direction in joint speaker diarization and separation. During this period, I developed a framework that integrated diarization and separation, incorporated diarization methods into separation, and combined both tasks in joint training. In addition, the proposed methods were developed to handle recordings with high and low speech overlap. The research resulted in a publication in IEEE Signal Processing Letters published in July 2025. Moreover, during this stay, I continued the work on a manuscript for the IEEE/ACM Transactions on Audio, Speech, and Language Processing, which was eventually published in August 2024. I participated in two projects that focused on improvements in speech separation and diarization, and information extraction with foundational audio models and large language models. Throughout all my visits, I had the opportunity to participate in the "Reading Group" sessions attended by CLSP doctoral students, where the latest developments in speech and natural language processing were discussed. I also attended CLSP seminars led by researchers from other universities and industry, which allowed me to establish professional contacts. At the end of my visit, I was invited to share my research findings and conclusions during one of these seminars.

1.5 Outline of the Thesis

The following chapters of this thesis describe the fundamentals and contributions of the research presented in Publications I, II, III, IV, and V.

Chapter 2 presents the state-of-the-art, related work and general fundamentals for the research described in this thesis in the areas of speaker recognition, speaker diarization, and joint speaker diarization and separation. Moreover, Section 2.3 of this Chapter explains the metrics used to evaluate the proposed systems, and Section 2.4 presents general descriptions of the datasets used in the experiments and highlights other datasets that are widely adopted in the domain.

The next Chapter 3 provides a concise overview of the work, contributions, and serves as an introduction to the series of publications. Section 3.1 outlines Publications I and II and their contributions to the field of speaker recognition. Section 3.2 details the developments of the non-autoregressive approach, with Subsection 3.2.1 focusing on its application to speaker diarization, as presented in Publications III and IV. Finally, Subsection 3.2.2 demonstrates the versatility of the non-autoregressive approach within the proposed framework for the joint speaker diarization and separation tasks.

Chapter 4 summarizes the thesis and contributions of the presented Publications, and draws out potential future work.

The end of this thesis concludes the Appendix, which contains full texts of five Publications that this thesis is based on.

Chapter 2

Research background

The purpose of this chapter is to present an overview of the background, fundamentals, and state-of-the-art relevant to the presented research. The review presented in the following sections does not aim to be exhaustive, but rather covers the works that are important and relative to the contributions presented in this thesis, as well as introduce the most important aspects of the investigated research areas. For a more in-depth discussion, the reader is referred to a comprehensive review of publications that examine these topics in greater detail, such as [9, 67, 116, 140].

2.1 Speaker recognition

The speaker recognition task plays an important role in the speech processing field. The methods developed in this domain are often the basis for other speaker characterization tasks, e.g. speaker diarization, emotion recognition, disease detection, etc. In speaker recognition, we can distinguish two distinctive tasks: speaker verification and speaker identification, both depicted in the general diagrams in Figure 2.1. The goal of speaker verification is to make a binary decision as to whether the person is who they claim to be, i.e., answering the question *"is it person X speaking?"*. Speaker identification aims to determine the identity of the speaker present in a particular recording, i.e., to answer the question *"who is speaking?"*. When speaker identification is limited to a predefined group of known (registered) speakers, the task is referred to as a *closed set* identification. When speaker identification allows speakers out of the registered group (i.e. unknown speakers), then the identification is from an *open set*. Speaker recognition systems can also be divided into *text-dependent* and *text-independent* systems. Text-dependent systems require the speaker to always use a specific phrase/password. In contrast, text-independent systems, which are investigated in this thesis, do not impose such a constraint.

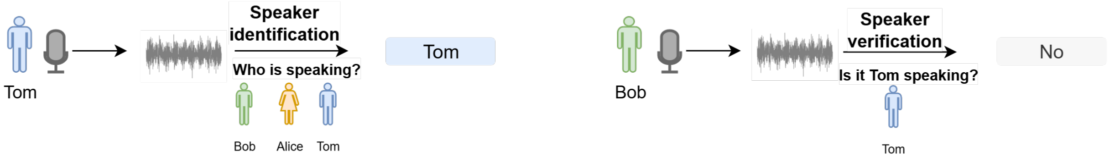


Figure 2.1: The general examples of the speech recognition tasks: identification and verification.

A more detailed diagram of the speaker verification process is presented in Figure 2.2. The process has two phases: enrollment and test. During the enrollment phase, a voice recording or recordings of a particular person are provided. Based on these, the reference speaker representation is computed. Next, in the test phase the test voice recording is provided from which in a similar way the speaker representation is extracted. Using the enrollment and test models, their similarity is evaluated in the backend (presented as the "Comparison" block in Figure 2.2). This stage produces a score/likelihood as to how models are similar, in order to assess whether they are from the same speaker. The obtained score is compared with an empirically selected threshold and returns a binary decision whether the voice samples are from the same speaker or not. Speaker identification task can be seen as N speaker verification operations: given a voice sample, the system performs comparisons with all N speaker models. The model that yields the highest score is selected as the identification result, provided that, in case of open-set identification, the score also exceeds a predefined threshold. For this reason, speaker recognition research focuses on the speaker verification task, as the development in verification can be easily scaled for the identification task. In this thesis, proposed methods were developed for the speaker verification task.

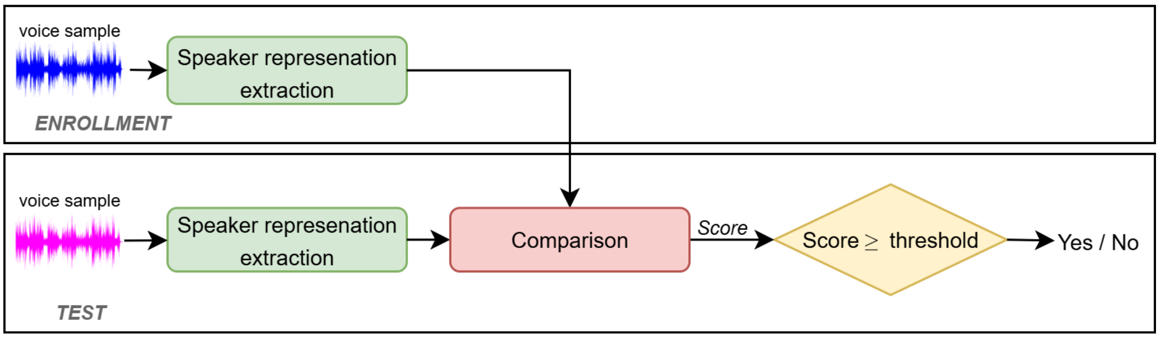


Figure 2.2: The general diagram of the speaker verification process.

2.1.1 Speaker verification system processing steps

The general processing of the speaker verification system can be divided into the following steps: preprocessing, feature extraction, speaker (representation) modeling, and backend scoring.

Preprocessing

The preprocessing step includes any algorithms that process the raw input signal. Usually the Voice Activity Detection (VAD) is applied in order to filter out non-speech time frames, and keep only speech fragments for further processing. The simplest and most commonly used VAD is the energy-based algorithm [145], which processes the recording frame by frame, computes the energy for each frame, compares it to a predefined threshold, and then evaluates the proportion of voiced and unvoiced frames within a contextual window. More detailed analysis of VAD techniques can be found in [112]. Signal preprocessing is also a stage in a speaker recognition system where methods such as speech enhancement, denoising, dereverberation, and speech separation can be employed to improve the robustness of speaker representation under adverse acoustic conditions.

Feature extraction

In most of the speaker verification system, preprocessed signal is followed by feature extraction step. In fact, feature extraction that is robust to adverse conditions and enhances the discriminative property of speaker modeling has also been an important research topic, presenting different variants [7, 34, 152, 71, 195, 142, 114]. For many years, the most well-established features have been Mel-Frequency Cepstral Coefficients (MFCC) [34]. Since MFCC extraction is a standard procedure for many speech processing tasks, its processing steps will be shortly outlined. The overall workflow is summarized in Figure 2.3. First, the signal undergoes pre-emphasis filtering to boost high frequencies. It is then divided into short, overlapping frames, typically 20 ms in length with a 10 ms stride, and multiplied by a window function such as Hamming or Hanning. This ensures the assumption of stationarity of the speech signal within each frame [1]. Each frame is then transformed using the Fast Fourier Transform (FFT) to compute its power spectrum. Next, triangular filters distributed along the Mel scale are applied. The Mel scale reflects human auditory perception, offering finer resolution at lower frequencies and coarser resolution at higher ones. Finally, a Discrete Cosine Transform (DCT) is applied to decorrelate the filterbank coefficients and produce a compact representation. With the development of DNN-based speaker modeling techniques, the coefficients of (log-)Mel-filterbanks (FBanks) emerged as a strong competitor to MFCC [9]. Mel-filterbank extraction follows the same steps as MFCC, but omits the DCT, and in some cases also the logarithmic operation.

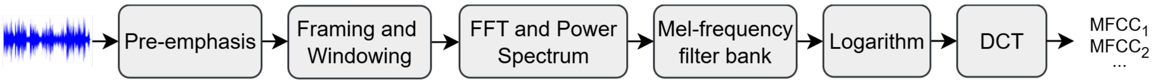


Figure 2.3: The general steps of the MFCC features extraction.

Speaker modeling

The core of the verification system is the speaker modeling part. Robust and discriminative speaker representation extraction/modeling is the area in the speaker recognition system where the research community mainly strives for improvement.

Since the late XX century, **Gaussian Mixture Models (GMMs)** have been the dominant method for speaker recognition for over two decades [116], first proposed in [153]. Shortly, the method models the speaker features with GMMs, i.e. mixture of Gaussian probability density functions distributions defined by a set of μ_g mean vectors, Σ_g covariance matrices, and π_g weights: $\Omega = \{\pi_g, \mu_g, \Sigma_g | g = 1,.., G\}$ for G Gaussians, also called components. Then, the probability of a single feature time frame ϕ_t can be obtained as:

$$f(\phi_t|\Omega) = \sum_{g=1}^{G} \pi_g \mathcal{N}(\phi_t|\mu_g, \Sigma_g), \tag{2.1}$$

and the final probability for the sequence $\Phi = \{\phi_1, ..., \phi_T\}$ can be formulated as:

$$p(\Phi|\Omega) = \prod_{t=1}^{T} f(\phi_t|\Omega). \tag{2.2}$$

In order to apply the method for speaker verification, two models are required: target speaker model and an alternate speaker model (representing non-target speakers). This allows the test data to be evaluated against both models, with the more likely one determining the accept/reject decision. The alternate speaker model led to the concept of the Universal Background Model (UBM) [154], which represents all speakers other than the target. In practice, the UBM is a large GMM trained to capture the speaker-independent distribution of speech features across a broad population, derived from multiple utterances of speakers, representing an "average" speaker [171]. Usually, GMMs are trained with an Expectation-Maximization (EM) algorithm [37]. In [155] the UBM was employed as initialization for adapting to the speaker-specific model with the enrollment recordings, which is also called as GMM-UBM method. The adaptation of the UBM to the recordings of a particular speaker was done via Maximum A Posteriori (MAP) adaptation [56]. The adapted speaker models were more effective and reliable than those trained independently.

The speaker verification system score can be computed as the difference of the log-likelihood ratios of the test recording with UBM (Ω_{UBM}) and speaker-specific model (Ω_s):

$$\Lambda(\Phi) = \log p(\Phi|\Omega_s) - \log p(\Phi|\Omega_{\text{UBM}}) \tag{2.3}$$

One of the problems with the presented method is the dependence of the final result on the recording length. That guided the research direction into fixed-dimensional representations of the recordings, which proved especially effective since they enable the use of machine learning classifiers. This led to the approach of the so-called *supervectors* from the GMMs, introduced for the first time in speaker recognition in [90]. Usually, the GMM supervector is represented with stacked vectors of GMM means of the recording (speaker) adapted UBM. This development enabled the application of different techniques, such as Support Vector Machines (SVMs) [31, 21, 20] and Factor Analysis (FA) [90], especially the Joint Factor Analysis (JFA) [89]. Since these methods are not employed in this thesis, detailed descriptions are omitted.

In general, the underlying assumption of the GMM-based approaches is that the GMMs representing an utterance are influenced by both speaker and channel variability. The general model representation can be formed as follows:

$$\mathbf{M} = \mathbf{m} + \mathbf{s} + \mathbf{c} \tag{2.4}$$

where **m** is a speaker/channel independent global mean (usually represented by UBM), **s** is a speaker part, and **c** represents channel variability. Throughout the years of using the approach, researchers have focused on modifications in the investigation of speaker and channel components [155, 17, 90, 89]. In JFA method, the supervector is typically represented with a linear combination of terms: (i) speaker/channel independent **m**, (ii) speaker-dependent with residual component $\mathbf{s} = \mathbf{V}\mathbf{y}_s + \mathbf{D}\mathbf{z}_s$ and (iii) channel-dependent $\mathbf{c} = \mathbf{U}\mathbf{x}$, which yields

$$\mathbf{M} = \mathbf{m} + \mathbf{V}\mathbf{y}_s + \mathbf{D}\mathbf{z}_s + \mathbf{U}\mathbf{x}. \tag{2.5}$$

The term (i) is usually represented by UBM and remains unchanged. The vectors $\mathbf{y}_s$, $\mathbf{z}_s$, $\mathbf{x}$ are standard normal vectors. The matrices $\mathbf{V}$, $\mathbf{D}$ $\mathbf{U}$ are low rank and capture the variability in the appropriate spaces. More detailed method description can be found in [91].

The next strong baseline, especially popular in the second decade of the XXI century, was the **i-vector** method [36, 35], superseding the GMMs. Recognizing that channel factors include speaker-dependent information, both speaker and channel factors were integrated into a single entity known as the total variability space. I-vectors simplified previous approaches to a model, which assumes to generate:

$$\mathbf{M} = \mathbf{m} + \mathbf{T}\mathbf{v} \tag{2.6}$$

where $\mathbf{m}$ is a global mean which is speaker and session independent, $\mathbf{T}$ is a model parameter, known as total variability matrix, $\mathbf{v}$ is the mentioned *i-vector* representation, also called as *identity* vector and known as *intermediate* representation as given its intermediate size between a supervector and an acoustic feature vector. The goal of $\mathbf{Tv}$ modeling is to capture both speaker and channel effects within a unified representation space, allowing channel variability to be mitigated directly. This contrasts with the earlier approaches, where speaker and channel characteristics were modeled as separate factors. Moreover, the lower dimensionality of the i-vectors, comparing to supervectors, allows for an application of different methods that can compensate the channel effects, which had been difficult to implement with the high-dimensional supervectors. In contrast to supervectors, which can have thousands of dimensions depending on the number of Gaussians and features (for example, for 2048 Gaussians and 19 features one obtains $2048 \times 19 = 38912$ supervector dimensions), i-vectors provide a compact representation of typically 100–600 dimensions.

With the growth of the dataset sizes and methods, research gradually shifted towards **deep neural network (DNN)** modeling. The development of DNN-based speaker modeling has undergone a transitional phase, during which hybrid DNN/i-vector approaches were explored. In these methods, DNNs were employed, for example, to estimate the UBM [102, 101], or to extract bottleneck features (BNFs) [65, 58], which either replaced or complemented traditional MFCC features. Approximately since the DNN-based era, speaker (vector) representations have also been called *speaker embeddings*, which was inspired from the speech recognition field, where words were transformed into *word embedding* representations. Since part of thesis contributions concern DNN-based speaker representation extraction, more detailed description of the approach is presented Chapter 2.1.2.

Backend

The extraction of the speaker representations is followed by a step that allows to compare the similarity between enrollment and test models and output a final decision. This stage may also incorporate techniques to compensate for channel-related differences in representations and facilitate domain adaptation.

Firstly, the **model normalization** can be applied as one of the possible channel compensation techniques that helps mitigate the impact of external speaker variability embedded in the model. The most common techniques are Within Class Covariance Normalization (WCCN) [68], Nuisance Attribute Projection (NAP) [22], Linear Discriminant Analysis (LDA) [10, 36]. Each of the approaches learns a projection that suppresses non-speaker variability. In [53] authors proposed to use whitening and length-normalization, in order to mitigate the non-Gaussian i-vector behaviour. The most commonly used, including experiments presented in the thesis Publications, is the LDA method with length-normalization. LDA is a popular approach to reduce the vector dimensionality. It allows for decreasing the size of the vector while preserving key information. Its goal is to estimate such vectors that are as close to each other as possible for the same speaker (minimize within-class variability) and as far apart as possible for different speakers (maximize between-class variability). In other words, the task of LDA is to project the vector into a space where classes are well separated.

Scoring is the main step in the speaker verification system backend. The most common scoring methods are Probabilistic Linear Discriminant Analysis (PLDA) [81, 146, 88] and cosine distance (also known as Cosine Distance Scoring - CDS). PLDA was widely used for i-vectors, helping to compensate for any adverse channel effects [171]. The method was proposed independently in [81] and [146]. In general, during times when the i-vector-based approach was dominating the field, the development of the speaker verification systems was accompanied by development and research into PLDA methods, proposing many variants of this technique. The most common PLDA approach models the distribution of the speaker representations, e.g. i-vector, as:

$$\mathbf{v} = \boldsymbol{\mu}_{\mathrm{v}} + \mathbf{S}\mathbf{z}_s + \boldsymbol{\epsilon}_{u,s}, \tag{2.7}$$

where $\boldsymbol{\mu}_{\mathrm{v}}$ is a global mean, $\mathbf{S}\mathbf{z}_s$ is a speaker-dependent component and $\boldsymbol{\epsilon}_{u,s}$ is a latent channel variable. The score with the PLDA method for the i-vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ is computed as a logarithm of the likelihood ratio in which $\mathbf{v}_1$ and $\mathbf{v}_2$ are from the same speakers versus $\mathbf{v}_1$ and $\mathbf{v}_2$ are from different speakers. The second popular scoring technique is the Cosine

Distance Scoring (CDS), which simply calculates the inner product between $\mathbf{v}_1$ and $\mathbf{v}_2$ and normalizes by their vector lengths:

$$CDS(\mathbf{v}_1, \mathbf{v}_2) = \frac{\mathbf{v}_1^{\mathrm{T}}\mathbf{v}_2}{||\mathbf{v}_1|| \cdot ||\mathbf{v}_2||}. \tag{2.8}$$

Score normalization is the next step that operates on the obtained scores and allows to standardize the score distribution [3, 115]. A speaker verification system produces two types of score distributions: target and non-target. The target distribution corresponds to cases where the enrollment and test utterances come from the same speaker, while the non-target distribution reflects comparisons between different speakers. Without normalization, the target and non-target score distributions may vary significantly across different enrolled speaker models. As a result, it becomes infeasible to define a single detection threshold that works consistently across all models. In the case of the same speaker model, score distributions can vary depending on the test utterance conditions, such as recording channel, acoustic environment, or language of the recorded speech. The normalization step adjusts the score distributions by shifting and scaling for individual models and/or conditions, enabling the use of a single detection threshold. These adjustments are typically estimated using a set of reference utterances, commonly referred to as the *normalization cohort*. Many techniques were developed that differ mainly in the way data is selected for the cohort [8, 155, 115, 210, 4, 154]. In general, the normalized score can be represented as:

$$score_{norm} = \frac{score - \mu_{norm}}{\sigma_{norm}}, \tag{2.9}$$

where μ_{norm} is a shift and σ_{norm} is a scaling factor. The μ_{norm} and σ_{norm} are usually calculated respectively as the mean and standard deviation of the scores of obtained the test or/and enrollment with the cohort files.

The last element of the system's backend is **calibration**. The performance of speaker verification systems is highly dependent on the training data domain, meaning that score distributions and optimal decision thresholds can vary across datasets. To address this, score calibration is applied to transform raw scores into interpretable likelihood ratios using a linear function [14, 15, 45].

2.1.2 Deep Neural Network-based speaker modeling

One of the first works on DNN-based embeddings is publication [184], which concerns text-dependent speaker recognition. In the training phase, the feed-forward model is trained for the speaker classification task, which means that the model learns to identify, i.e. classify speakers from the training dataset. During the test phase, the output after the activation function of the last hidden layer is averaged on the temporal axis and the speaker representation, called *d-vector*, is obtained. The d-vector method assumes that the embedding space trained on a training set can reliably generalize to unseen speakers during evaluation. A key characteristic of the d-vector architecture is that the feed-forward structure processes the input utterance only at the frame level. The next work [70] proposed a variant based on recurrent layers and proposed an end-to-end system that simultaneously learns speaker embeddings and a similarity metric to compare embedding pairs.

The mentioned approaches were an inspiration for the next important structure presented in [176, 173], which introduced segment-level processing for the text-independent speaker verification system. The structure gained a property of processing recordings with variable length by introducing a temporal statistics pooling layer inside of the structure. The pooling layer aims to aggregate information from frame-level processing by computing the mean and standard deviation over the temporal axis, concatenating these statistics, and passing the resulting representation to subsequent layers.

The next development is the core structure for DNN-based speaker verification models, where speaker embeddings called *x-vectors* [175] are extracted. The model architecture is based on previous work [173]. However, in this study, the authors demonstrated that data augmentation significantly improves x-vector performance, achieving results that exceeded state-of-the-art systems of that time. The general structure of the x-vector model is presented in Figure 2.4. It can be separated into part that is processing the recording on the frame-level and the part which processes the recording on the segment level. The first is composed of five Time Delay Neural Network (TDNN) layers [141] which as input take features extracted from the recording. TDNN layers are equivalent to 1-D convolutional layers applied along the time dimension, often with a dilation larger than one [116]. Information from the frame-level part is fed to the segment-level part, which first aggregates and compresses it into fixed-size input with statistics pooling. The pooling layer computes the mean and standard deviation along the time axis, concatenates both vectors and forwards to two feed-forward layers. On the top of the structure, a linear layer with a number of outputs equal to the number of speakers in the training dataset is applied, followed by a softmax function. The x-vectors are extracted from the first feed-forward layer after statistics pooling. The model is trained for the speaker classification

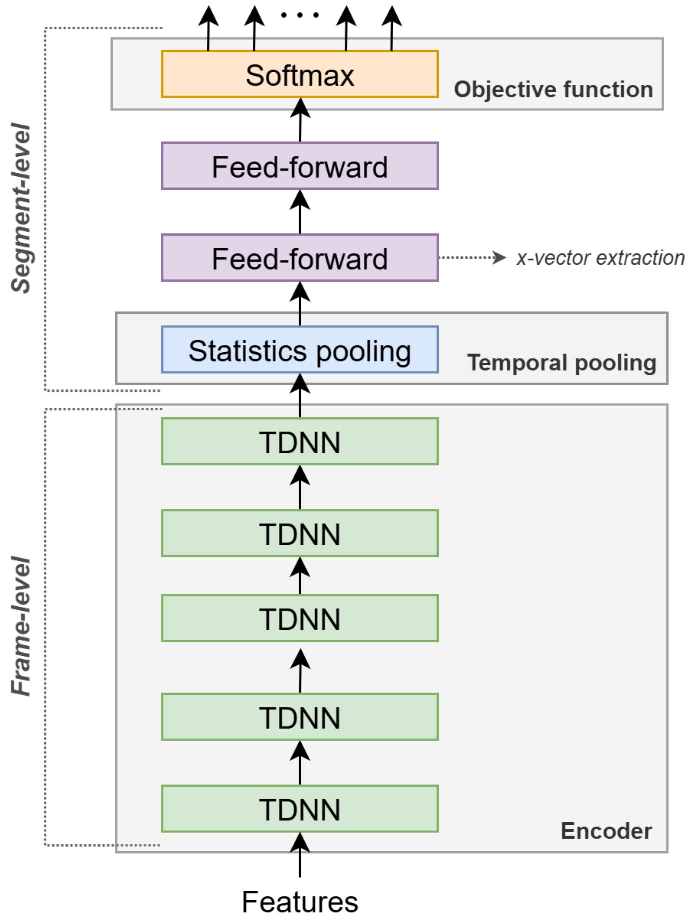


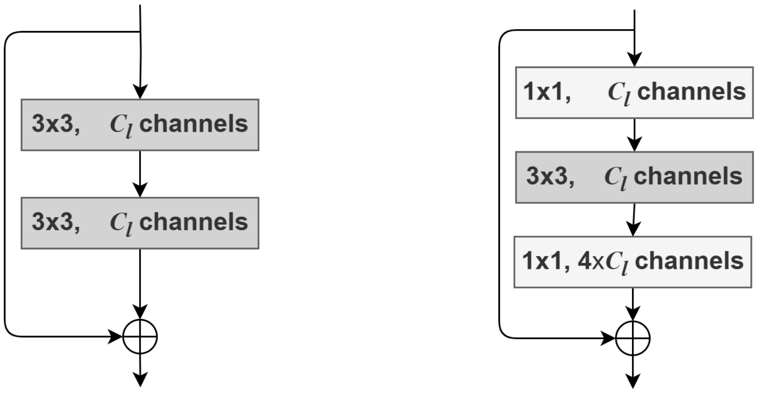
Figure 2.4: The general diagram of x-vector structure.

task, whose goal is to indicate which speaker from the training set is present in the input recording.

The proposal of the x-vector model highlighted specific components within the DNN speaker recognition structure as key areas of interest, which have since been extensively investigated by the research community. All three of these components are marked in Figure 2.4. The first concerns the proposals for the structure of the frame-level (encoder) part. The second area concerns the temporal pooling and methods to effectively transform information from frame- to segment-level. The last part is the objective function and its possible optimizations and enhancement of discrimination capabilities.

Encoder modifications

The x-vector structure has been developed to other popular alternatives. The first worth mentioning is Extended TDNN (ETDNN) [174] which enlarges the frame-level part by adding a linear layer between every TDNN layer and increasing the temporal context of the frame-level part. The other development is the Factorized TDNN (FTDNN) [144]. In this model, the weights of each frame-level layer are factorized into two low-rank matrices and one of them is forced to be semiorthogonal in order to avoid loss of information.

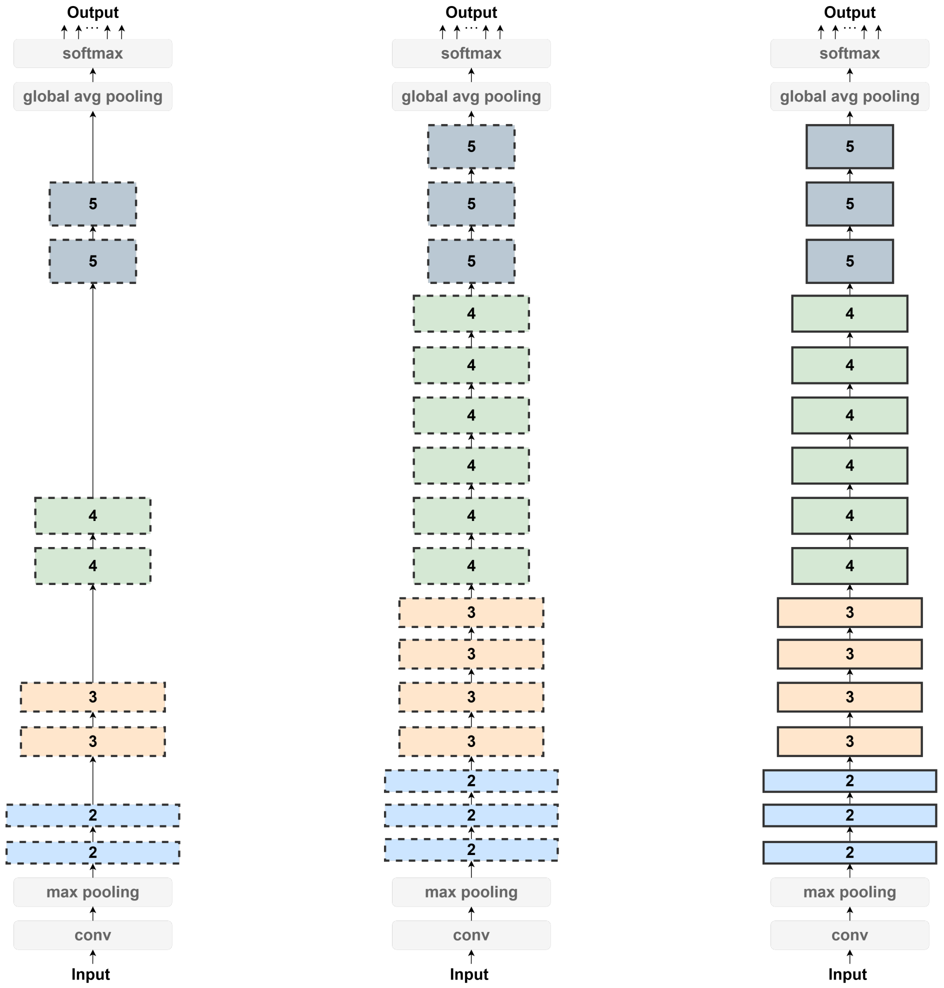


(a) Basic residual block.

(b) Bottleneck residual block.

Figure 2.5: Diagram of (a) basic and (b) bottleneck residual blocks of the ResNet structure [69]. C_l indicates base number of channels.

Soon after the x-vector proposal, the ResNet [69] structure has become another more and more popular encoder option for the DNN model for speaker verification. The ResNet architecture is built of 2D convolutional layers with residual connections. These connections are characteristic of the ResNet structure and were proposed to prevent vanishing gradients. Since one of the contributions of the thesis includes proposed DNN structures that improve over ResNet-based encoders, the structure is explained in more detail next. The core building blocks of ResNet are residual blocks of two types: basic and bottleneck, illustrated in Figure 2.5. Both types of blocks contain the so-called shortcut/residual connection, which means that the input is added to the block output. If the dimensions of the block input and output are identical, the shortcut connection is simply an identity mapping, allowing the input to be directly added in an element-wise manner to the block output. If the dimensions of the block input and output are different (e.g. due to a change in the channel dimension or downsampling), the linear projection (usually implemented as 1×1 convolutional layer) is additionally applied to the input to match the output dimensions and enable element-wise addition. The basic



(a) ResNet-18 (b) ResNet-34 (c) ResNet-50

Figure 2.6: Diagrams of (a) ResNet-18, (b) ResNet-34, (b) ResNet-50 structures. Numbers inside blocks denote its level, the blocks with solid lines represent the bottleneck type, while dotted represent basic residual.

block is composed of two 3 × 3 convolutional layers with C_l base number of channels. The bottleneck block is composed of three convolutional layers with kernels: 1 × 1, 3 × 3, 1 × 1 and C_l base number of channels. Note that the last convolutional layer of the bottleneck block has increased the number of channels by a factor of four. C_l number depends on the block level, and each level 2, 3, 4, 5 has receptively 64, 128, 256 and 512 number of base channels. Figure 2.6 presents three selected types of ResNet: ResNet-18, ResNet-34, ResNet-50. The suffix number of the ResNet name indicates the number of layers in the structure, excluding the pooling layers from this counting. Blocks with solid lines represent the bottleneck type, while dotted represent basic residual. The numbers inside blocks denote their level, which indicates the number of base channels and how downsampled is the feature map with respect to the input. It should be noted that, in contrast to the architecture described later in Section 3.1, the relationship between the block level l and the downsampling factor in these structures follows 2^l, while in Section 3.1 it follows 2^{l-2}. This difference arises from the fact that the original ResNet architecture includes two downsampling operations (each by a factor of 2) in the initial *max pooling* and convolutional (*conv*) layers, which are omitted in the version proposed for the speaker verification task in Section 3.1. To ensure consistency in the block-level notation with Publication II, this discrepancy is intentionally preserved in the present section.

The speaker recognition community has widely adapted the residual network for the task, introducing different modifications, either using the structure almost intact [30, 27] or adjusting it for speaker verification, e.g., using the typical for this task temporal pooling methods [18, 19, 186, 177, 77, 185]. In [205] the authors referred to the embedding extracted from the ResNet structure as the *r-vector*. In [136] authors inspired by the ResNet architecture introduced residual connections between TDNN layers, in order to gain a wider context with a deeper architecture. In the context of ResNet-based speaker verification systems, the ECAPA-TDNN model (Emphasized Channel Attention, Propagation, and Aggregation) [39] is worth highlighting, which has emerged as a strong state-of-the-art approach by effectively integrating concepts from both x-vector and ResNet-based architectures.

Proposals for temporal pooling

One of the integral parts of the x-vector-based structure is a pooling layer which estimates mean and standard deviation statistics. Several works tried to improve on that. The use of statistics pooling brings the assumption that each time frame contributes equally to speaker information [9], which may not be true, e.g. when particular frames contain less speech and more noise or silence than the others. One of the modifications was the introduction of self-attention-based layers that help assign higher weights to frames that contribute more to the final result [137, 209, 192]. In the paper [19], the authors proposed a Learnable Dictionary Encoding (LDE) pooling layer. The method employs two sets of learnable parameters: dictionary component centers and their associated weights. For each frame-level embedding, i.e. embeddings before the pooling operation, the distances to all component centers are computed. These distances are then converted into weights using a softmax function with an additional learnable scaling parameter called the smoothing factor. Using the resulting weights and distances, a weighted average is calculated for each component center, producing pooled representations that summarize the frame-level features. Other pooling methods explored in the literature are, e.g., NetVLAD [199], GhostVLAD [199], Spatial Pyramid Pooling [207] and others. The contributions of the thesis do not concern the development over temporal pooling, thus, for a more detailed review, the reader is referred to [9].

Angular-based objective functions

The final element of interest in speaker verification architectures is the objective function. In the literature, the combination of the cross-entropy loss and the output layer with the softmax function is often called a softmax loss, which will also be used in this thesis. In the following section, we discuss the modification of softmax loss within the context of speaker recognition treated as a multi-class classification task. Among the various alternative objective functions proposed, the angular softmax losses have gained significant popularity and demonstrated strong effectiveness, which is further explained in this subsection.

The standard softmax loss is effective in maximizing the separation between classes, but lacks the ability to directly minimize within-class variability. To address this, a margin parameter is introduced, which enforces tighter clustering of features within the same class by reducing the within-class distance between the input representations and their corresponding class weights [9]. First, let me briefly introduce the basics of the angular-based losses. The general cross-entropy loss function for the speaker classification problem can be defined as follows [9]:

$$\mathscr{L} = -\frac{1}{N}\sum_{i=1}^{N} \log P_{y_i} \tag{2.10}$$

with $i = 1, 2, ..., N$, and N being the minibatch size and y_i being the ground-truth label of a training example. The output of the last layer with the softmax function can be presented as:

$$P_{y_i} = \frac{e^{\mathbf{w}_{y_i}^T \mathbf{x}_i}}{e^{\mathbf{w}_{y_i}^T \mathbf{x}_i} + \sum\limits_{k=1, k\neq y_i}^{K} e^{\mathbf{w}_k^T \mathbf{x}_i}} \tag{2.11}$$

$\mathbf{w}_k^T \mathbf{x}_i$ is a dot product between the last layer weights $\mathbf{w}_k$ for the kth class, where $k = 1, 2, ..., K$ and K is the number of speakers to classify that is equal to the number of speakers in the training set, and input vector for the ith minibatch example $\mathbf{x}_i$ with $i = 1, 2, ..., N$, N being the minibatch size. The dot product operation can be equivalently rewritten as a product of vector norms and the cosine angle between them:

$$\mathbf{w}_k^T \mathbf{x}_i = \|\mathbf{w}_k\| \, \|\mathbf{x}_i\| \cos(\theta_{i,k}). \tag{2.12}$$

Next, the class weights are normalized $\|\mathbf{w}_k\| = 1$. For the sake of further explanations the $\|\mathbf{x}_i\|$ is replaced with a scale variable $s(\theta_{y_i})$ and $\psi(\theta_{y_i})$ replaces the cosine of the angle for the ground-truth label. After the aforementioned modifications, equation (2.11) can be rewritten as:

$$P_{y_i} = \frac{e^{s(\theta_{y_i})\,\psi(\theta_{y_i})}}{e^{s(\theta_{y_i})\,\psi(\theta_{y_i})} + \sum\limits_{k=1,k\neq y_i}^{K} e^{s(\theta_{y_i})\,\cos(\theta_{i,k})}} \tag{2.13}$$

The angular softmax losses introduce modifications mainly in $\psi(\theta_{y_i})$. The Angular Softmax (AS) proposed in [107] was incorporated for the speaker task in works like [78, 136, 19]. Its proposal is to modify $\psi(\theta_{y_i})$ by introducing margin m_{AS}, where $m_{\mathrm{AS}} \geq 2$, i.e.

$$\psi(\theta_{y_i}) = \cos(m_{\mathrm{AS}}\theta_{y_i}). \tag{2.14}$$

In [107], the authors explain the introduction of a margin by considering an example of a two-class classification problem. Assuming the input belongs to class 1, it will be classified as such if its softmax posterior probability is higher than that of class 2, i.e. $P_1 > P_2$, which yields the requirement that:

$$\cos(\theta_1) > \cos(\theta_2) \quad \rightarrow \quad \theta_1 < \theta_2. \tag{2.15}$$

In order to make the requirement more strict, the margin parameter is introduced as in (2.14), which yields

$$\cos(m_{\mathrm{AS}}\theta_1) > \cos(\theta_2) \quad \rightarrow \quad \theta_1 < \frac{\theta_2}{m_{\mathrm{AS}}}. \tag{2.16}$$

Presented requirements put a constrain on $\theta_{y_i} \in [0, \frac{\pi}{m_{\mathrm{AS}}}]$. In order to generalize it and facilitate its optimization into a monotonically decreasing function, it is redefined as:

$$\psi(\theta_{y_i}) = (-1)^{\gamma}\cos(m_{\mathrm{AS}}\theta_{y_i}) - 2\gamma, \tag{2.17}$$

where $\theta_{y_i} \in [\frac{\gamma\pi}{m_{\mathrm{AS}}}, \frac{(\gamma+1)\pi}{m_{\mathrm{AS}}}]$, $\gamma \in [0, m_{\mathrm{AS}} - 1]$, $m_{\mathrm{AS}} \geq 1$. In the AS method, the scale function is kept unchanged $s(\theta_{y_i}) = \|\mathbf{x}_i\|$.

The Additive Margin Softmax (AMS) [190] modifies the angular function to

$$\psi(\theta_{y_i}) = \cos(\theta_{y_i}) - m_{\text{AMS}}, \tag{2.18}$$

which normalizes $\|\mathbf{x}_i\| = 1$ and sets $s(\theta_{y_i}) = s_{\text{AMS}}$ with some fixed value. AMS method was used for the speaker recognition task, e.g. in [54, 199, 203].

The Additive Angular Margin Softmax (AAS) [38] is currently the most popular angular softmax for the speaker verification task. The method introduces

$$\psi(\theta_{y_i}) = \cos(\theta_{y_i} + m_{\text{AAS}}), \tag{2.19}$$

normalizes $\|\mathbf{x}_i\| = 1$ and sets $s(\theta_{y_i}) = s_{\text{AAS}}$ which is an arbitrarily selected value. AAS was used in several speaker verification research publications, e.g. [39, 187, 197]. In general, angular-based softmax losses optimize the angular distribution of feature representations, enabling the network to generate embeddings that can be effectively evaluated using a simple cosine similarity-based backend.

Lastly, another interesting approach was presented in [208], where Adaptively Scaling Cosine Logits (AdaCos) was proposed for the face recognition task. The method introduces adaptation of the scale function $s(\theta_{y_i})$ during the network training. Although this approach has not been applied to speaker recognition, its description is important as it serves as the foundation for one of the contributions discussed later in this work. The authors observed that both scale and margin can modulate the supervision strength by controlling the prediction probability P_{y_i}, where stronger supervision improves class separability. The proposed approach adapts only the scale value, while the angular function $\psi(\theta_{y_i})$ is kept intact. The goal of the derived adaptation function was to select $s(\theta_{y_i})$ that makes the predicted probability P_{y_i} highly sensitive to changes in the angle θ_{y_i}, by finding the point where the gradient of P_{y_i} with respect to θ_{y_i} is maximized (i.e., where its second derivative equals zero).

2.2 Speaker diarization

Speaker diarization is a task that answers the question "who spoke when". This means that diarization identifies segments where the same speaker is present, while also detecting overlapping speech between speakers as well as regions of silence. Figure 2.7 presents an auxiliary diagram of this task. Diarization may be perceived as a task similar to speaker recognition. However, speaker information for both systems has different character -

speaker recognition directly points/verifies the absolute speaker identity, while for the diarization task it is enough to discriminate between speakers within the recording and recognize relative differences between speakers, without specifying the speaker's identity.

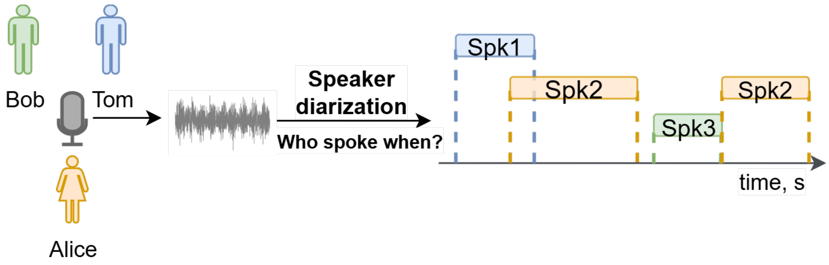


Figure 2.7: The general scheme of the diarization task.

2.2.1 Cluster-based diarization

For a long time, speaker recognition models were a core part of diarization systems, as part of the so-called cluster-based approach [169, 55, 41]. The steps of its processing flow are presented in Figure 2.8. The cluster-based diarization can also be classified as a modular based approach as it combines a few independent processing modules. The input recording is initially processed by Voice Activity Detection (VAD), which may be optionally preceded by a preprocessing step. VAD ensures that only speech segments are retained for further processing, while the optional preprocessing addresses signal-level challenges such as noise and reverberation. The next step is speech segmentation, which in the simplest version is dividing speech into overlapping chunks. From each chunk, a speaker embedding is extracted using a speaker model extractor. These embeddings are then compared and scored against each other to determine the similarity between chunks. Finally, based on the scoring result, speaker embeddings from all chunks are clustered, to obtain the answer which embeddings belong to the same speaker, that is in which chunks the same speaker is speaking. Some systems may also include additional postprocessing steps, such as resegmentation or overlap detection mechanisms.

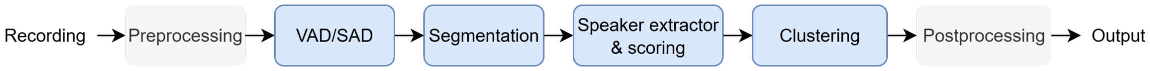


Figure 2.8: The general scheme of the processing flow of the cluster-based diarization system.

Voice Activity Detection

Voice Activity Detection (VAD), also called Speech Activity Detection (SAD), is a module that detects speech in order to filter out the non-speech regions. In general, VAD is a module which is incorporated in several speech tasks, including ASR, speaker recognition, and others, so its development is independent to the speaker diarization task. The challenge for the VAD system is domain adaptation, when system trained for one scenario, can perform worse in another one. The domain difference include not only various acoustic conditions, but also characteristic of the conversations (proportions of speech, silence, and overlap), which is different for telephone conversations, dinner-party scenarios or for household (smart home applications) recordings. Thus, VAD is usually adjusted, trained or either fine-tuned for a particular use case. VAD and its potential errors (assigning silence to speakers or discarding speech segments) heavily impact the performance of the entire diarization system. For this reason, the diarization systems are often reported in the literature with oracle VAD decision, where the ground-truth information is used instead of the real VAD system. Despite these challenges, currently VAD modules can perform reasonably well [99]. Various solutions have been developed starting from energy or spectrum-based [33, 2, 82], statistical-based [29, 183], machine learning [43, 51] to DNN-based methods [106, 79, 57].

Segmentation

The next step, segmentation, may take the form of either uniform segmentation or Speaker Change Detection (SCD) module. The goal of SCD is to segment audio into partitions, where the segment boundaries are dictated by detected speaker change points. Typically, the module divides audio into short chunks, compares similarity between them, and determines whether they belong to the same speaker based on these similarity scores [9]. SCD modules have been explored in the literature [24, 170, 13, 5]. However, since it may produce segments of varying lengths, this poses a challenge for newer speaker embedding extractors such as i-vectors [36] or DNN-based approaches [184, 175], in which inconsistent input duration may yield inconsistent representations. Thus, for these extractors, the most commonly used is the uniform segmentation [140, 55, 191, 104]. It divides the recording into fixed-size segments, usually 1.5 seconds long with a 0.75 second overlap. In uniform segmentation, the time window must be short enough to maintain the assumption of only one speaker within each segment, yet long enough to allow the extraction of high-quality speaker representations. Uniform segmentation is currently the dominant approach in cluster-based speaker diarization systems.

Speaker representation extraction and scoring

Segmented chunks are then fed into the pre-trained speaker representation extractor, e.g. i-vector [165] or x-vector [167], where one representation is generated for each chunk. The chunk-level representations are compared with each other to obtain similarity scores using techniques such as PLDA [165] or CDS [191]. The techniques for speaker representation extraction, as well as scoring, overlap with advances for the speaker recognition task, which has already been described in Chapter 2.1.

Clustering

The last step is to cluster and decide which chunks can be assigned to the same speaker. It means that the goal of the clustering is to group representations, where each speaker is represented by one cluster group. Although there is a wide interest in research of possible improvements in this area [119, 206, 200], the best performing and robust are classical clustering algorithms: spectral clustering [168, 125, 191, 105] and especially Agglomerative Hierarchical Clustering (AHC) [167, 55, 6, 135].

In this step it is important to mention Variational Bayesian Hidden Markov Model, especially a variant designed for x-vectors [41, 98] called VBx. The approach is based on Hidden Markov Model, where each state represents a speaker with possible transitions between speakers, including a probability of not changing the state. The model's parameters are initialized with pretrained PLDA. An important property of this approach is that it yields posterior probabilities of presence in a particular chunk for each of the speakers, which can be effectively leveraged for overlap assignment.

Postprocessing

The postprocessing stage encompasses several optional refinements to improve the output of speaker diarization systems. One common technique is resegmentation, which aims to fine-tune segment boundaries for increased accuracy [166, 40]. Postprocessing may also involve fusion of diarization outputs from multiple systems [80, 12], with notable approaches such as DOVER (Diarization Output Voting Error Reduction) leveraging voting-based strategies to combine hypotheses [179, 151, 198].

Traditional clustering-based diarization systems typically assume single-speaker segments, making them inherently incapable of handling overlapping speech. To address this limitation, specialized overlap handling methods have been developed, generally involving a two-stage process of (1) overlap detection, followed by (2) speaker assignment to the detected overlap regions [16, 150].

Cluster-based diarization constitutes a dominant approach to diarization. However, recent advances in this field include end-to-end approaches. The main contributions of this thesis concern a purely end-to-end approach, which are described in Chapter to follow. Thus, for further details on the stage-wise approach, please refer to [140].

2.2.2 End-to-end speaker diarization

The problems of the aforementioned cluster-based diarization method are a few. The modularity of this approach means that each of the components is optimized separately and not directly for the diarization result. Moreover, by design, cluster-based approaches do not inherently handle overlapping speech segments; therefore, additional mechanisms must be incorporated to address overlaps effectively. The end-to-end neural network-based speaker diarization (EEND) framework [47] was a solution to these problems. Figure 2.9 presents the general diagram of the framework. The input into the structure is the sequence of features $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$ of length T and dimension F. Features are then processed by encoder, which produces the frame-level embedding sequence $\mathbf{e} = \{\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_T\}$, where embeddings are D-dimensional. Next, the embeddings are forwarded to the model's backend. In the first proposal [47] it was composed of a simple linear layer with a sigmoid function, with K number of outputs, which is equal to the assumed maximum number of speakers. The output is producing the posterior probabilities of the speaker k present in the time frame t, that is, $\widetilde{\mathbf{y}}_k = \{y_{k,1}, y_{k,2}, \ldots, y_{k,T}, \}$, where $k = 1, ..., K$. Each output layer corresponds to a separate speaker track. By applying a threshold, these probabilities can be converted into binary decisions indicating speaker presence or absence. This simple mechanism allows the model to identify overlapping speech when multiple outputs are set to 1 for the same time frame, or silence when all outputs are equal to 0. The training objective is a well-known binary cross-entropy loss. The order of speaker activities returned by the diarization model may differ from the order in the ground-truth answers. However, this does not indicate an error in the diarization results, as speaker identities are inherently permutation-invariant. To resolve the ambiguity in assigning output tracks to the correct ground-truth labels, permutation invariant training (PIT) [201] scheme is employed, which is inspired by the speech separation field. In PIT, the loss between the diarization result and each possible order permutation of the reference labels is computed, and the one that results in the lowest value is selected as the loss result.

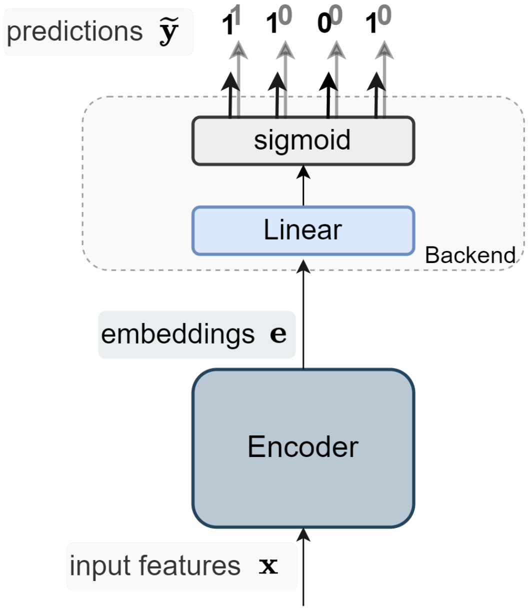


Figure 2.9: The general diagram of the EEND framework.

As mentioned in the beginning of this subsection, a key advantage of EEND is its inherent ability to handle overlapping speech, whereas cluster-based approaches require additional mechanisms to manage overlap effectively. The end-to-end approach allows to set a diarization result as an objective, thus optimizing the network directly for the diarization result. An additional advantage is that the model can be easily fine-tuned for different domains [140].

The initial EEND architecture employed an encoder built with bidirectional Long Short-Term Memory (BLSTM) layers [47]. However, it was soon replaced by a self-attention-based architecture utilizing Transformer encoder layers [48], offering improved performance and modeling capabilities. Some other works also proposed improvements in the encoder part, i.e. in [108, 103] Conformer [66] was adopted in place of self-attention layers. With the presence of residual connections between self-attention layers in EEND, some studies have investigated the use of auxiliary losses computed after each layer [202, 49].

One of the limitations of the presented versions of the EEND was the limited number of speakers that the system can handle, which is constrained by the architecture and number of classification outputs of the last layer. Speaker-wise conditional EEND (SC-EEND) [50]

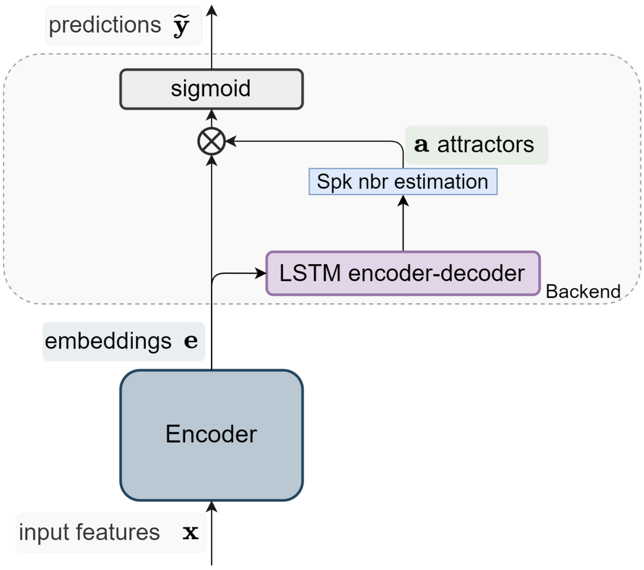


Figure 2.10: The general diagram of the EEND-EDA framework.

was one of the works designed to tackle variable number of speakers in the recording. The diarization output probabilities are decoded through a chain rule, where the diarization output is estimated sequentially speaker-by-speaker, conditioned on the activity from the previous estimations. This process continues as long as the diarization output indicates no speech.

Another important proposal that allows processing a flexible number of speakers is EEND with encoder-decoder based attractors (EDA) [73]. Given EEND-EDA effectiveness, the SC-EEND has not gained a lot of attention in the research community. The general EEND-EDA diagram is presented in Figure 2.10. In EEND-EDA the encoder embeddings are fed into the model's backend which is EDA composed of the two LSTM layers connected in the encoder-decoder manner and a linear layer with a single output. The EDA task is to produce the so-called attractors - vector representations of the speakers present in the processed recording. The LSTM encoder-decoder can in theory produce an infinite number of attractors. In order to limit and detect the actual number of attractors needed, i.e. number of speakers in the recording, the linear layer is applied one-by-one on each produced attractor. When the probability returned by this layer drops below a threshold, the attractor generation is stopped. Note that the mechanism performed by the linear layer represents a speaker counting function in the model. Next, the obtained attractors are used to compute the dot product with the frame-level embeddings. The result is processed by a sigmoid function, which gives a diarization posterior probability and can be presented in the following manner:

$$\widetilde{y}_{s,t} = \sigma(\mathbf{e}_t^{\mathrm{T}} \mathbf{a}_s) \,, \tag{2.20}$$

where $\mathbf{e}_t$ is a frame-level embedding at time t and $\mathbf{a}_s$ denotes the attractor for the s-th speaker.

In general, the EEND approach has received a lot of attention, also with the main development focus on its backend evolution. It is worth mentioning that EEND with global and local attractors, also known as EEND-GLA [74, 75], developed from EEND-EDA to deal with a larger number of speakers than those present during training. The embedding sequence obtained from EEND is split into smaller chunks. The chunks are then processed by the EDA backend to produce (local) attractors and diarization results, with the assumption that in a short chunk the number of speakers is relatively small. To compute inter-chunk dependence among the local attractors, a transformer decoder is employed, treating the local attractors as queries and the frame embeddings as keys and values that convert to representations that can be clustered. Clustering is applied to produce the final diarization output by grouping attractors corresponding to the same speaker, thereby merging diarized segments that belong to that speaker. The attractors produced with an original EEND-EDA approach, i.e. when attractors extracted directly based on the whole recording, are referred to as global attractors. During inference of the EEND-GLA, the model at first uses global attractors to produce the diarization result. If at this step the number of detected speakers is equal to or is higher than the maximum number of speakers seen during training (set to four in the original papers), then the local attractors are used to estimate the final result. Otherwise, the result obtained from the global attractors is used directly.

It is also worth noting studies that explore the integration of EEND with a cluster-based approach [95, 94], the so-called EEND-vector clustering (EEND-VC). The EEND-VC framework combines the advantages of EEND and cluster-based diarization systems, where EEND provides a precise and overlap-aware diarization output for short chunks, and clustering combines these results to enable processing of long recordings. The model adopts the traditional EEND approach with a linear layer on top [48], but modifies the architecture to output not only the diarization decisions, but also speaker representations per each diarization track. The system follows the assumption that the chunk processed by EEND is short enough to contain no more than 2 or 3 speakers. After processing all chunks from the recording, a global decision is made by clustering speaker representations and their associated diarization activities across chunks. This clustering is performed with the constraint that speaker representations originating from the same chunk are not merged, ensuring consistent and accurate speaker attribution across the entire recording. Additional speaker loss is applied to encourage generation of speaker discriminative representations. In the next work [93] authors proposed to incorporate trainable unfolded infinite Gaussian mixture model as a clustering step, enabling the joint training of both EEND and clustering. In this approach, the parameters are estimated using variational Bayes inference, which yields improved performance on the recordings with a larger number of speakers. In a further work on EEND-VC [96] authors directed attention to properly splitting of the recordings into chunks. The authors highlight that the EEND-VC system imposes a rigid constraint on the number of speakers per chunk. While longer chunks can improve diarization accuracy, they may violate the assumption of a maximum of 2 or 3 speakers per chunk. Additionally, fixed-length segmentation complicates integration with the downstream tasks such as ASR, where segment boundaries may not align with utterance semantics. It can also degrade the quality of diarization by introducing unnaturally short or poorly aligned segments. To address this issue, [96] proposes the Graph-PIT-EEND-VC framework, which eliminates fixed segmentation in favor of an utterance-by-utterance processing approach inspired by source separation. The framework leverages Graph-PIT [189] training and instead of the typical diarization objective function, the encoder is trained as a two-channel VAD, which detects speech activity while assigning overlapping speakers to the separate channels. Instead of enforcing fixed chunks, the model uses a more flexible representation of utterance activity, assuming a maximum of two overlapping speakers at any time but allowing an unlimited number of speakers overall.

2.2.3 Speaker diarization and separation

In this section a review of important research on a combination of speaker diarization and separation is presented. Speech separation is a specific form of source separation that aims to produce isolated audio tracks for each speaker present in a mixed recording. A closely related task is target-speaker extraction, in which a single speaker is isolated from the mixture using an auxiliary cue such as a reference audio sample, spatial information, or descriptive metadata. Another related concept is speech enhancement, which seeks to improve the quality and intelligibility of a speech signal by reducing noise, interference, and reverberation. By default, enhancement is applied on recordings with only one speaker, while in case of separation or extraction the recordings typically contain more than one speaker. In the literature terms *speech separation* and *speaker separation* have been used interchangeably, therefore, the same convention is adopted in this thesis. Its general scheme is presented in Figure 2.11.

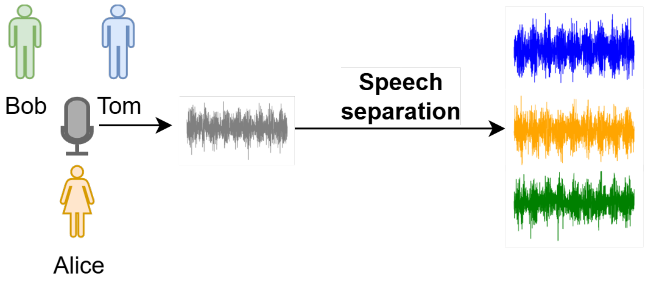


Figure 2.11: The general scheme of the speaker separation task.

The diarization and speech separation are, in fact, closely related tasks which differ in the granularity of the answer: diarization returns a binary decision about speaker activity, while separation usually uses the time-frequency speech activity masks to extract speaker's speech. In the literature, they were also investigated as tasks that can complement each other's challenges and provide mutual support.

An in-depth analysis was presented in [149] where the authors assemble the modular model with separately trained separation, diarization, and speech recognition (in this order) components for multi-talker speech recognition, with the aim to design a system which answers the "who said what and when?" question and study the impact of each step on the final system performance. In the speech separation guided diarization (SSGD) approach [44] authors combine separation for speaker embedding-based diarization to deal with overlapping segments that contaminate speaker representations. As described in Section 2.2.1, one of the disadvantages of the cluster-based diarization system is the general assumption that only one speaker is present in the chunk, which introduces the problem of dealing with the speech overlap. The authors propose to incorporate diarization results from two systems: (1) cluster-based diarization and (2) speech separation combined with VAD. The results from both paths are compared to obtain the relative diarization error rate between these two systems. The formula is similar to standard diarization metric Diarization Error Rate (DER) (metric explained in the next subsection), but as ground truth the cluster-based diarization result is used. If the relative DER is lower than the selected threshold, the answer from the separation part is used. If it is higher, the result from the diarization part is used. The alternative strategy to the described process is to fine-tune the speech separation model with speaker representations from the clustering system. In the latter scenario, the results from the fine-tuned separation model alone are sufficient. The effectiveness of the solution was demonstrated by its first-place finish in the DIHARD-III challenge, which is a renowned challenge focused on the speaker diarization task, including the recordings ‘in the wild’, of various conditions and multiple speakers in the challenging scenarios. In a more recent work [182], the authors propose speaker separation via neural diarization (SSND). Here, the diarization is used directly to guide speaker separation. EEND-EDA [73] is used to obtain the speaker boundaries along with their embedding representations. This information guides the speaker separation system by helping to assign speaker waveforms into a two-channel output — ensuring that overlapping speakers are placed in separate channels. This approach enables the effective use of speaker separation models even for long recordings. In addition, it eliminates the need for a stitching process, which improves the efficiency of separation computations. At the same time, it enables the stitching of utterances which belong to the same speaker, enhancing the overall coherence of the separation output. In the context of the research proposed in this thesis, the usage of EDA from EEND-EDA diarization for the separation-only model – SepEDA [178] is important to mention. The EDA attractor mechanism has been injected into SepFormer [180] model, which enabled the separation model to deal with recordings that contain a flexible and unknown number of speakers.

Recently, more and more interest has been brought to joint modeling of diarization and separation to solve both tasks simultaneously. In [111] the authors propose Joint End-to-End Neural Speaker Diarization and Separation (EEND-SS). The model incorporates ConvTasNet separation [110] and EEND-EDA diarization [73] trained jointly. The bottleneck features of the separation model are concatenated with the input diarization features, and the diarization speaker counting is used for the separation structure to choose the number of masks for separation. Target-Speaker based Separation (TS-SEP) [11] is the model for joint diarization and separation developed by extending the diarization system which is Target-Speaker Voice Activity Detection (TS-VAD) [118]. TS-VAD is an important diarization approach that presented great performance as a part of the winning system in the CHIME-6 challenge [193]. It can be considered as a DNN-based system combining the joint VAD, segmentation, and speaker identification model, applied on top of the cluster-based diarization system. TS-VAD model inputs are the i-vectors of the speakers present and the sequence of acoustic features of the recording. In order to obtain i-vectors, the model first applies the cluster-based diarization to obtain rough speaker representations. Then, TS-VAD iteratively estimates the diarization result with i-vectors from the previous iteration and refines the i-vector estimation using the obtained diarization result. The weakness of the solution is its limited maximum number of speakers that it can handle, which is predefined by the model's architecture. In the TS-SEP separation-diarization system, the TS-VAD model is trained first. As soon as TS-VAD obtains reasonable diarization performance, the last binary diarization layer is modified to a two-dimensional linear layer with the ability to produce time-frequency masks. In the next step, the model is trained for the separation task. The diarization result is then retrieved directly from the masks by computing the mean value of the frequency bins and applying additional smoothing on top of the results. Another recently proposed approach is PixIT [85] which is based on the Dual-Path Recurrent Neural Network (DPRNN) separation model [109]. It concatenates the pre-trained WavLM features [25] with separation features from the separation encoder. The structure is also modified by adding in parallel to the separation decoder a small diarization decoder composed of linear layers that outputs the diarization result.

2.3 Evaluation metrics

This part of the chapter presents and explains the main metrics used to evaluate the particular systems addressed in this thesis.

2.3.1 Speaker verification

As described in the beginning of this chapter, the output of a speaker verification system is a score that assesses how likely/similar test model is to the enrollment model. The score compared to the threshold returns a binary decision. If the score is higher than or equal to the set threshold value, then the system decides that the enrollment and test are from the same speaker. If the score value is lower, they are treated as coming from different speakers. Speaker verification model and its decisions are evaluated with the *trial list* which defines the enrollment and test pairs, along with *target* or *non-target* (alternatively referred to as *impostor*) labels, which indicates whether the pair is from the same speaker or, in the latter case, from different speakers.

The scores obtained from the speaker verification system for the target and impostor trials can be plotted in the form of distributions, presented in the conceptual illustration in Figure 2.12. The decisions obtained by applying a score threshold are compared with the labels in the trial list. The wrong assignment can emerge into two types of errors: (1) False Rejection (FR), also known as a False Negative, when *target* is classified as *non-target* and (2) False Positive (FP), also known as a False Acceptance, when *non-target* is classified as

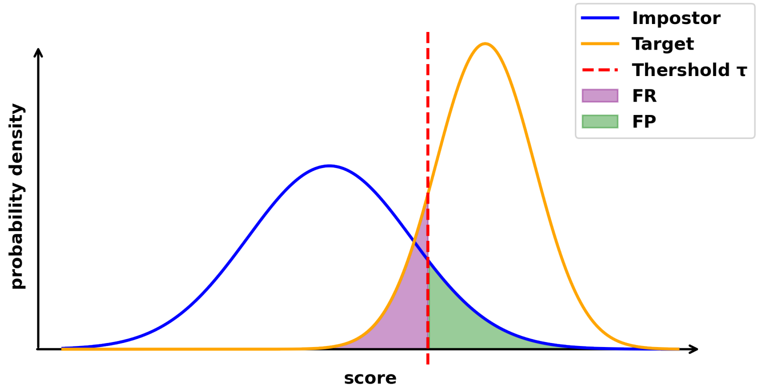


Figure 2.12: Illustrative distributions of the impostor and target scores.

target. Notice that the decision of accepting/rejecting a trial pair depends on the threshold value τ, thus the number of False Positive/False Rejection decisions depends on τ (also depicted in Figure 2.12). This leads to formulating the False Positive Ratio (FPR) and the False Rejection Ratio (FRR) as:

$$\mathrm{FPR}(\tau) = \frac{N_{\mathrm{FP}}(\tau)}{N_{\mathrm{non-target}}} \cdot 100\%, \tag{2.21}$$

$$\mathrm{FRR}(\tau) = \frac{N_{\mathrm{FR}}(\tau)}{N_{\mathrm{target}}} \cdot 100\%, \tag{2.22}$$

where $N_{\mathrm{FP}}(\tau)$ and $N_{\mathrm{FR}}(\tau)$ is the number of False Positive and False Negative decisions for the particular τ, and $N_{\mathrm{non-target}}$ and N_{target} is the number of non-target and target trial pairs.

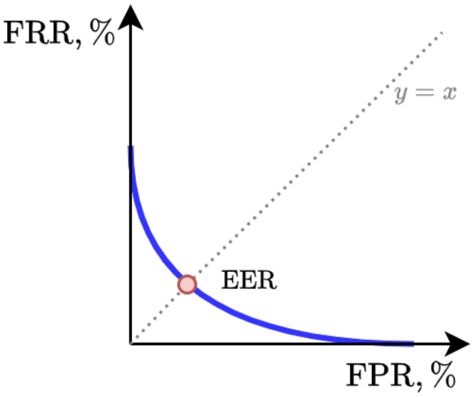


Figure 2.13: Illustrative diagram of DET plot.

By changing the threshold values τ, the relation of $\mathrm{FPR}(\tau)$ and $\mathrm{FRR}(\tau)$ can be obtained. Plotting the FRR with respect to FPR presents the so-called Detection Error Tradeoff (DET) plot [113], presented in Figure 2.13. The diagram presents the most commonly used metric to evaluate speaker verification system known as the Equal Error Rate (EER). It is the error value, where, for an estimated threshold τ_{EER}, the False Positive Ratio and False Rejection Ratio values are equal, i.e. $\mathrm{FPR}(\tau_{\mathrm{EER}}) = \mathrm{FRR}(\tau_{\mathrm{EER}}) = \mathrm{EER}$. It is worth noting that increasing τ value can increase FRR error and decrease FPR, which improves the general safety of the speaker verification system application. The other way around, decreasing τ value can decrease FRR error and increase FPR, which can make system more convenient for the genuine user at the cost of the system's safety. In practical applications, operating a speaker verification system at the threshold corresponding to the EER may not be the optimal choice, thus the τ value is selected based on the particular use case.

In order to assess system performance at the chosen operating point, adjusted to the specific application of the system, the second metric is often used, which is the Detection Cost Function (DCF) [126], introduced and commonly used in NIST evaluations. The metric applies the weights, i.e. costs/penalties for particular errors: C_{FR} - cost of FR, C_{FP} - cost of FP:

$$\mathrm{DCF}(\tau) = P_{\mathrm{tar}} C_{\mathrm{FR}} \mathrm{FRR}(\tau) + (1 - P_{\mathrm{tar}}) C_{\mathrm{FP}} \mathrm{FPR}(\tau), \tag{2.23}$$

where P_{tar} is the predefined probability of the target trial, which can be interpreted as an indication of how often the target speaker attempt can be present at the system input. In the NIST SRE 2008 evaluation, the weight values were selected as $C_{\mathrm{FR}} = 10$, $C_{\mathrm{FP}} = 1$ and $P_{\mathrm{tar}} = 0.01$, which means that the speaker verification system is punished tenfold for rejecting the target speaker than for accepting the impostor. For example, in a real-world scenario, when identifying a known criminal's voice from evidence recordings, it may be preferable to accept some false positives—such as investigating an innocent speaker—rather than risk missing the target speaker entirely and failing to detect the criminal [67]. In the research community, these parameters are most commonly set as $C_{\mathrm{FR}} = C_{\mathrm{FP}} = 1$ and $P_{\mathrm{tar}} = 0.01$. In the literature and in this thesis, the reported value often is the minimum value of the $\mathrm{DCF}(\tau)$ function, called minDCF:

$$\mathrm{minDCF} = \operatorname*{argmin}_{\tau} \mathrm{DCF}(\tau). \tag{2.24}$$

In case of both EER and minDCF, the lower the value, the better the system is.

2.3.2 Speaker diarization

To measure speaker diarization performance, the Diarization Error Rate (DER) is the most widely used metric [46, 140]. The DER is a sum of three different errors:

- FA - amount of False Alarm speech, i.e. silence labeled as speech or amount of time of overlapping speech attributed to more speakers than actually present;
- Miss - missed detection of speech or amount of time of overlapping speech attributed to fewer speakers than actually present;
- SC - confusion between speakers, i.e. amount of time where speech is assigned to the wrong speaker.

The sum is divided by the overall duration of the speech in the reference, counting the overlaps as well (*Duration*):

$$\mathrm{DER} = \frac{\mathrm{FA} + \mathrm{Miss} + \mathrm{SC}}{Duration}. \tag{2.25}$$

In order to compute the DER, the speaker tracks of the diarization system need to be assigned to the ground-truth labels. The one-to-one mapping is done with the Hungarian algorithm [97], which relies on finding the optimal assignment by solving the corresponding bipartite graph matching problem. In [46], the so-called collar was introduced, which is a region around the boundaries of the reference segments that is not used for metric computation. Typically, its value is set to 0.25 seconds. It was proposed in order to account for inconsistencies in human annotations and errors. This tolerance helps to reduce penalization for minor timing mismatches. In general, a lower value of the DER metric indicates a better-performing diarization system.

Another diarization evaluation metric is the Jaccard Error Rate (JER), proposed in the DIHARD II challenge [159]. Although JER highly correlates with DER, it gives equal weight to all speakers in the recording regardless of whether they contribute largely to the total amount of speech or not. This is relevant in situations where dominant speakers occur, but the presence of all speakers is equally important. In order to obtain the JER value, for each speaker, the False Alarm (FA) and Miss errors (Miss) are computed and then divided by the union of the speaking time of the speaker's ground truth and hypothesis (*Total*):

$$\mathrm{JER} = \frac{1}{N}\sum_{i=1}^{N} \frac{\mathrm{FA}_i + \mathrm{Miss}_i}{Total_i}, \tag{2.26}$$

where N is the number of speakers in the ground-truth answer. Since JER uses the union of the reference and hypothesis, its value never exceeds 100%, while DER can surpass

100%. JER tends to be higher than DER when a subset of the speakers dominates the audio recording. Similarly to DER, the lower the value of JER, the better the system.

2.3.3 Speaker separation

The quality of speaker separation systems can be assessed with many different metrics, e.g. Perceptual Evaluation of Speech Quality (PESQ) [156], Short-Time Objective Intelligibility (STOI) [181], Signal-to-Distortion Ratio (SDR) [188] or even by applying the ASR system on the separated output and measuring the Word Error Rate (WER) value [28].

In Publication V presented in this thesis, the Scale-Invariant Signal-to-Distortion Ratio (SI-SDR), or more specifically SI-SDR improvement (SI-SDRi) [157] is selected as a metric to evaluate the speech separation system. Scale-invariant indicates that the metric is insensitive to the amplitude scale of the signal and compares only the shapes of the target and estimated waveforms. The metric estimates the ratio between the clean, target signal and the interference or distortion introduced during the separation process, which can be defined as

$$\text{SI-SDR} = 10\log\frac{|\alpha s|^2}{|\alpha s - \hat{s}|^2} \text{ for } \alpha = \underset{\alpha}{\operatorname{argmin}} |\alpha s - \hat{s}|^2 \tag{2.27}$$

where for the optimal target case

$$\alpha = \frac{\hat{s}^T s}{||s||^2}, \tag{2.28}$$

where s represents the ground-truth signal and $\hat{s}$ is the estimated one. The improvement of the SI-SDR (SI-SDRi) refers to the difference between the SI-SDR of the estimated signal to the target signal and the SI-SDR of the mixture to the target signal. In general, the higher the value of the metric, the better the separation achieved.

2.4 Datasets

This section presents the most relevant datasets for each task, as well as those used in the experimental evaluation of the proposed research.

2.4.1 Speaker recognition

NIST SRE datasets

In the context of the text-independent speaker recognition task, it is crucial to mention the datasets published during NIST SRE evaluations. Since 1996 the National Institute for Standards in Technology (NIST) has hosted many Speaker Recognition Evaluation (SRE) competitions every one or two years, the goal of which is to benchmark and drive speaker technology development. The Linguistic Data Consortium (LDC) collaborates with NIST to collect and provide data. Each SRE evaluation the NIST provides with a new and challenging dataset which, after the evaluation is finished, is released. The evaluations and provided data are designed with the aim to challenge the systems while ensuring constructive outcomes and conclusions and allowing to focus on a specific evaluation problem without introducing unnecessary complications. The amount of data is also selected to provide significant results and allow the participation of groups with limited resources [64]. Earlier SRE datasets are frequently included in training data, which is then used to build models for subsequent evaluations. In general, NIST evaluations contain a variety of data, primarily presented conversational telephony speech (CTS) recorded over public switched telephone networks (PSTN), supplemented over time with varied conditions such as languages, microphones, environments, demographics, and later expanded to include Voice over Internet Protocol (VoIP) and audio from video (AfV).

In the Publications presented as this thesis contribution, the NIST SRE has not been incorporated for speaker recognition experiments, however, some of the evaluation datasets were used for speaker diarization system training, namely the NIST Speaker Recognition Evaluation 2004, 2005, 2006, 2008 [127, 129, 128, 131, 134, 130, 132, 133], representing telephone speech at a sampling frequency of 8 kHz.

SITW

Speakers in the Wild (SITW) [117] is a hand-annotated database that features recordings of public figures collected from open-source media, allowing to test performance of the speaker recognition systems. It includes a total of 299 speakers, with on average eight recordings per speaker, and comprises both single- and multi-speaker audio segments. The novelty of the proposed database was the large number of speakers and recordings captured ‘in the wild’, featuring real reverberation, noise, compression artifacts, and within-speaker variability across sessions.

VoxCeleb

VoxCeleb datasets [122], namely VoxCeleb1 [123] and VoxCeleb2 [30] are the most commonly used and publicly available audio-visual datasets for benchmarking speaker recognition systems. The datasets collect the recordings of celebrities ‘in the wild’ of different nationalities, ages, languages, and professions, extracted from YouTube platform videos. The recordings were obtained with a fully-automated pipeline resulting in 352 hours of 1251 speakers for VoxCeleb1 and 2242 hours from 6112 speakers for VoxCeleb2. The dataset was an answer for the need for a dataset large enough to train and develop DNN based methods. Until then, the NIST SRE datasets have been sufficiently large for speaker recognition research. However, since they were not made publicly available, accessibility to the broader research community was and still is limited. The most common procedure is to use VoxCeleb2 for training, while VoxCeleb1 is used evaluation. Both VoxCeleb1 and VoxCeleb2 introduced train and test splits, along with trial lists. VoxCeleb2 [30], introduced after VoxCeleb1, proposed also three trial lists, that are commonly used to benchmark speaker recognition systems: **VoxCeleb1-O** (Original), which is the original trial list of the test part VoxCeleb1 dataset consisting of 40 speakers and 37611 enroll-test pairs (trials); **VoxCeleb1-E** (Entire), which covers the entire VoxCeleb1 dataset, and extends the Original in order to validate the speaker recognition systems with a dataset with a large speaker number in order to limit the possibility of overfitting to a small set; it consists of 1251 speakers and 579818 trials; **VoxCeleb1-H** (Hard) which also covers the entire VoxCeleb1 dataset but makes the list more challenging by limiting trial pairs to be from the same nationality and gender; it results in 550894 trial pairs from 1251 speakers.

2.4.2 Speaker diarization

CALLHOME (LDC2001S97)

CALLHOME [147] is one of the most popular benchmarks for the speaker diarization task that includes real telephone conversations of multilingual speech. It is a speaker segmentation dataset from 2000 NIST SRE, which contains approximately 17 hours of recordings from 500 sessions, with 2-7 speakers, usually with two leading the discussion.

DIHARD

DIHARD datasets are the collections of the recordings used to evaluate during DIHARD challenges, namely DIHARD I [158], DIHARD II [160] and DIHARD III [161]. Depending on the dataset, in general they cover mainly speech recordings at the sampling frequency of 16 kHz, single and multichannel, in the wild, mostly English and Mandarin. The goal of DIHARD evaluations is the development of diarization systems in the challenging, real-life conditions. DIHARD I introduced recordings from diverse domains such as YouTube videos, clinical or radio interviews, restaurant conversations, and audiobooks; DIHARD II added multichannel recordings; and DIHARD III expanded the dataset with additional telephone conversations.

AMI

AMI (Augmented Multiparty Interactions) dataset [23] contains multi-modal meeting recordings of 170 meeting sessions with 100 hours in total, with 3 to 5 speakers. It contains the recordings and data from the meetings in the office environments obtained with different types of devices: close-talking microphones (headset and lapel), far-field microphones (2 linear microphone arrays, 8 microphones each), individual and room-view video cameras, visual content from the slide projector and electronic whiteboard, and output from individual pens used by participants to take notes.

NIST SRE and Switchboard simulated mixtures and conversations

This paragraph does not focus on a specific dataset, but rather on an approach to generate large-scale simulated training data for EEND-based diarization systems. This is an important aspect to highlight, as such simulation techniques are crucial for effective training large neural models in the absence of extensive annotated real-world data. EEND-based models require large amounts of training data, typically thousands of hours, which cannot be feasibly obtained using traditional hand-annotated datasets, as they are generally too limited in size. Thus, the proper simulation procedure was introduced in [47] and was followed by many other researchers as a procedure to generate a large-scale simulated dataset for training of EEND models, where the recordings used were from Switchboard-2 (Phase I, II, III) [59, 61, 60], Switchboard Cellular (Part 1, Part 2) [62, 63], and NIST Speaker Recognition Evaluation datasets (2004, 2005, 2006, 2008) [127, 129, 128, 131, 134, 130, 132, 133]. The combined dataset results in recordings from 6381 speakers, all being telephone speech at a sampling frequency of 8 kHz. The simulation algorithm produces a multi-speaker audio mixture with background noise. It begins by sampling a set of speakers and, for each, selects a random number of utterances. These utterances are spaced using random time intervals, convolved with room impulse responses, and then concatenated. Each speaker's track is padded to match the longest one. All tracks are summed to form the mixture. Finally, background noise is added and scaled based on a randomly chosen signal-to-noise ratio (SNR) to produce the final mixed audio signal. As this procedure is widely adopted in the research literature, [100] pointed out that the simulated mixtures do not resemble real-life conversations, which can hinder the possible performance of the diarization model. To address this issue, the authors proposed a method for generating simulated conversations (as opposed to simple simulated mixtures). By extracting statistics from datasets that contain real conversations and applying them to simulate more natural conversational patterns, the approach leads to improved model performance. This method reduces the gap between models trained on simulated data and those applied to real-life scenarios, thereby minimizing the need for additional fine-tuning on real-world data.

2.4.3 Speaker diarization and separation

LibriMix and SparseLibriMix

LibriMix [32] is a well-known simulated dataset for speaker separation task, available in two versions: with two speaker recordings (Libri2Mix) and three speaker recordings (Libri3Mix). It was proposed as an alternative to existing separation benchmarks in order to address the problem of the limited number of available evaluation datasets. LibriMix contains mixtures simulated from LibriSpeech recordings [138], either clean or with added ambient noise from the WHAM! dataset [194]. LibriSpeech is a very popular dataset, used mainly for speech recognition systems, containing read audiobook speech. WHAM! includes noise samples which were collected in realistic environments with natural background noise, like bars, coffee shops, restaurants. Libri2Mix contains subsets: *train-360* with 212 hours, *train-100* with 50 hours, *dev* with 11 hours and *test* with 11 hours. Libri3Mix contains *train-360* with 146 hours, *train-100* with 40 hours, *dev* with 11 hours and *test* with 11 hours. The training sets *train-360* and *train-100* correspond to the two LibriSpeech training set sizes, containing approximately 360 hours and 100 hours of audio, respectively.

LibriMix represents a corpus with high overlap mixtures. In order to complement the LibriMix dataset with conversational-like data, the authors also proposed an alternative of SparseLibriMix (2 and 3-speaker variant), with recordings of a varying amount of speech overlap, i.e. 0%, 20%, 40%, 60%, 80%, and 100%. For SparseLibriMix the authors provided scripts to generate clean and noisy version only for the test set, as the length of noise samples from the WHAM! set is insufficient to generate the proper training set. Respectively, both test sets of SparseLibri2Mix (2-speaker version) and SparseLibri3Mix (3-speaker version) sum up to 6 hours. It is important to note that in the 3-speaker variant, the reported amount of overlap refers specifically to segments where all three speakers are talking simultaneously. This means that the actual amount of speech overlap, defined as at least two speakers speaking at the same time, is higher than the reported 3-speaker overlap alone.

LibriheavyMix

LibriheavyMix [84] is a recently proposed dataset that contains large amount (around 20 000 hours) of simulated recordings, tailored for speech separation, recognition, and diarization tasks (i.e. task of "who spoke what and when"). The introduced dataset includes a large volume of data and enhances the simulated recordings with added reverberation, multiple speaker turns to better reflect real-life conversational dynamics, and transcriptions enriched with proper punctuation, casing, and contextual information. Recordings contain from 1-4 speakers for the training set and 2-4 speakers for the development and test sets.

LibriCSS

LibriCSS [28] is a dataset proposed for continuous speech separation (CSS). CSS stands for separation of speech signals from the audio stream that may contain multiple speakers, different levels of overlap, and have durations of several hours. The authors of LibriCSS proposed to create conversations by combining utterances from the LibriSpeech dataset [138] and playing and recording them in real rooms. The recordings form 10 one-hour sessions, where each contains 10-minute "mini sessions", with different overlap: 0%, 10%, 20%, 30%, and 40%. The 0% subset is presented in two versions: with short (0.1-0.5 seconds) and long (2.9-3.0 seconds) silence segments applied between utterances in the simulated mixture. The dataset contains multi-channel recordings, and each session contains 8 speakers. LibriCSS provides the conversational-like mixtures that include both overlap and non-overlap regions, in contrast to other datasets for speech separation that contain only fully-overlapped recordings.

Chapter 3

Overview of conducted research and main contributions

This chapter presents the major contributions and research findings of this thesis in speaker recognition, speaker diarization, and the joint diarization and separation tasks. Detailed descriptions and results are presented in the articles, the full text of which is presented in the Appendix of this thesis.

3.1 Discriminative speaker representations

Publications I and II present contributions to the domain of DNN-based speaker recognition. As described in Chapter 2, at the time of this research work was performed, the speaker recognition community devoted attention to the development and understanding of the different parts of the DNN structure of the speaker recognition system, i.a. the objective functions and how to train models to generate discriminative representations. At the time of DNN-based speaker system developments, angular-based losses were becoming popular, with currently the Additive Angular Softmax being the well-established and widely adopted objective function.

The problem of the Additive Angular Softmax and similar loss functions is that their hyperparameters such as scale and margin are fixed. They have been derived empirically and adopted which is confirmed by numerous repetitions of their citation in various publications. Introduction of fixed parameters, not adjusted to the training dataset, may cause not optimal model training, and search for a proper set of parameters would require repeated model training, which is extremely time-consuming and ineffective. In Publication I this problem is solved by a proposal of two methods of a proper margin-only adaptation and the simultaneous adaptation of the margin and scale parameters that adjust to the size of the training dataset and adapt during DNN training in order to enhance the discriminative capabilities of the trained model.

Firstly, the adaptation procedure for the margin parameter was formulated. The following angular function was considered:

$$\psi_{\text{MAda}}(\theta_{y_i}) = \cos(\theta_{y_i} + m_{\text{Ada}}), \tag{3.1}$$

where m_{Ada} is a margin parameter whose value is adjusted during training, and θ_{y_i} being the angle between the input vector of the last classification layer and the vector weights of the y_i class, where y_i is a ground-truth label of the particular sample. The expression for softmax function P_{y_i} was given in equation (2.13) and is restated below for clarity:

$$P_{y_i} = \frac{e^{f_{y_i}}}{e^{f_{y_i}} + \sum\limits_{k=1,k\neq y_i}^{K} e^{f_{i,k}}}, \tag{3.2}$$

where f_{y_i} and $f_{i,k}$ are defined as $f_{y_i} = s(\theta_{y_i})\psi(\theta_{y_i})$ and $f_{i,k} = s(\theta_{y_i})\cos(\theta_{i,k})$, respectively. The goal is to set m_{Ada} such that it maximally affects the P_{y_i} curve with respect to θ_{y_i}. The optimal value of m_{Ada} can be determined by finding the maximum value of the absolute gradient of P_{y_i}, which can be obtained by finding the point where its second-order derivative is equal to zero. It results in the following update of the margin:

$$m_{\text{Ada}} = \arccos\left(\frac{1}{s_m}\log(B_{\text{MAda}})\right) - \Theta\,, \tag{3.3}$$

$$B_{\text{MAda}} = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1,k\neq y_i}^{K} e^{s_m\cos(\theta_{i,k})}, \tag{3.4}$$

where $\Theta = \text{median}(\theta_{y_1}, \theta_{y_2}, ..., \theta_{y_N})$ is the median over the angles for ground-truth examples in the processed minibatch, and s_m is a fixed scale parameter. B_{MAda} is a sum over exponential logit functions for non-ground-truth classes, for a batch of size N and total number of classes K in the training dataset. This adaptive margin method is referred to as MAda.

The presented margin adaptation scheme leads to formulation of the full parameter adaptation (called as ParAda) of the softmax-based cross-entropy loss function, which is a proper combination of adaptive scale and margin. The philosophy of ParAda is to gradually increase the training supervision and make the training more strict with the

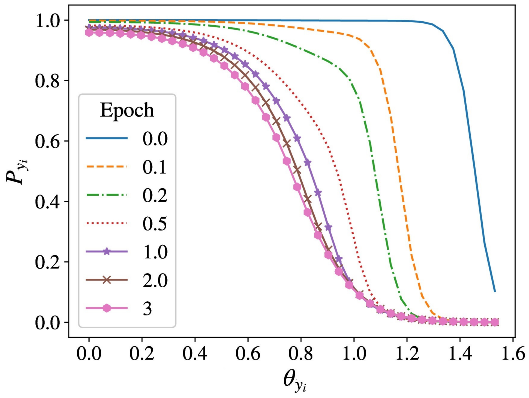


Figure 3.1: Probability curves $P_{y_i}(\theta_{y_i})$ at different epochs for presented ParAda function. Plot borrowed from Publication I.

network convergence, by giving more weight to margin adaptation, and with later epochs balance it with scale adaptation. It is achieved by modifying the shape of P_{y_i} plot and its inflection point during the different training stages by adapting the scale and margin parameters. This behaviour can be presented with the diagram in Figure 3.1. In the first epochs, the margin adaptation shifts the curve's inflection point toward high angle values, allowing the probabilities to vary more in this region. This is consistent with the fact that the ground truth angles θ_{y_i} are also high at this stage, when the model is still in its early training phase and its predictions are less reliable. As training progresses, the angles decrease in value, so that the curve is shifted towards the middle point of $\frac{\pi}{4}$ and the weight of the impact is moved from the adaptive margin to the adaptive scale, which continuously aims to make training more and more strict. This process is formulated in the following equations:

$$f_{y_i} = \lambda \cdot s_m \psi_{\text{MAda}}(\theta_{y_i}) + (1-\lambda) \cdot s_{\text{Ada}}(\theta_{y_i}) \cos(\theta_{y_i}), \tag{3.5}$$

$$f_{i,k} = \lambda \cdot s_m \cos(\theta_{i,k}) + (1-\lambda) \cdot s_{\text{Ada}}(\theta_{y_i}) \cos(\theta_{i,k}), \tag{3.6}$$

where s_{Ada} is the adaptive scale computed with AdaCos method [208], mentioned in Chapter 2. The weighing parameter $\lambda(m_{\text{Ada}})$ is also adaptive and adjusts according to the current margin values:

$$\lambda(m_{\text{Ada}}) = [1 + e^{a \cdot (m_{\text{Ada}} - b)}]^{-1}. \tag{3.7}$$

The hyperparameters a and b are selected empirically. The presented method not only solves the problem of nonoptimal adaptive angular loss parameters, but also provides faster network convergence and prime performance.

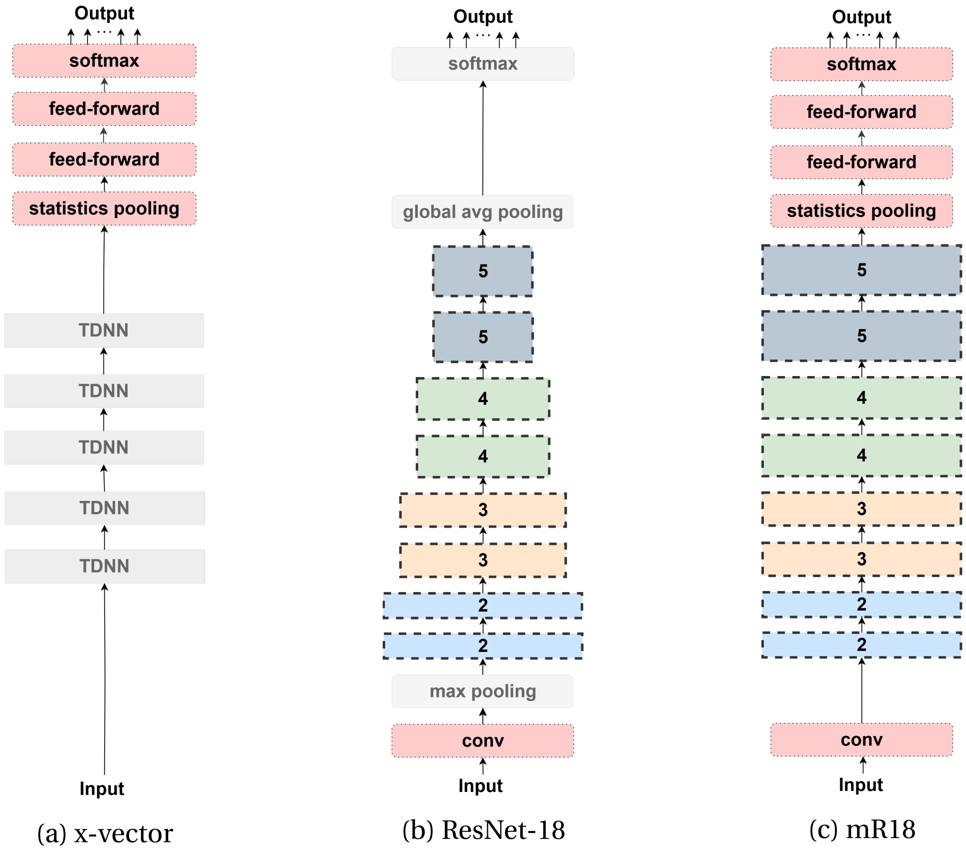


Figure 3.2: The diagram of the structures of (a) TDNN (x-vector) [175], (b) ResNet-18 [69] and (c) mR18 models. The red dashed line points the modifications introduced for mR18 structure.

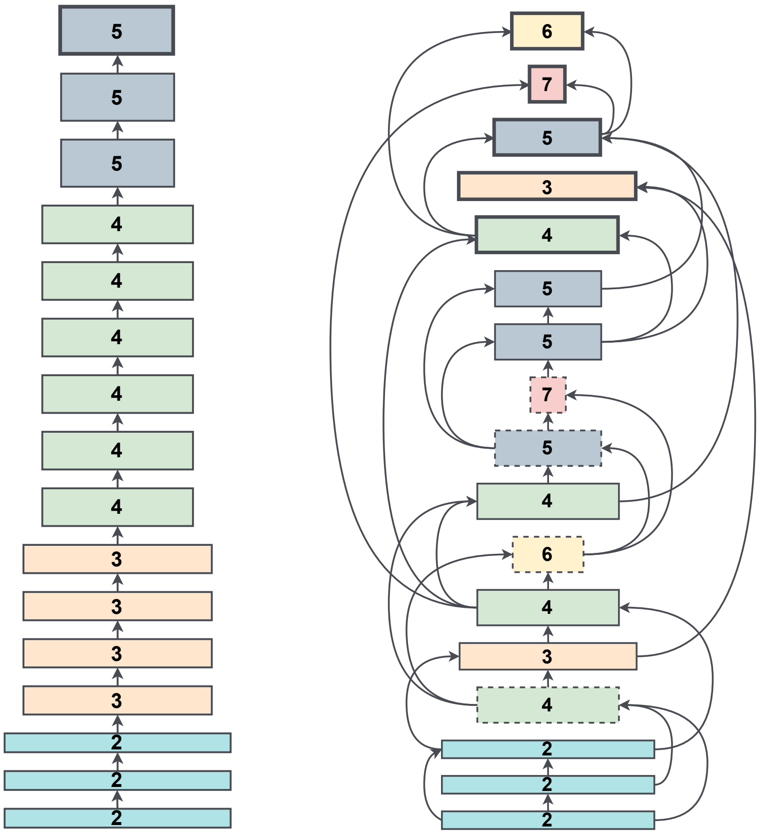
Another contribution introduced in Publication I is a modification of the ResNet structure. At that time ResNet structures just started to gain attention of the research community as a good modeling structure for the speaker recognition task. In this work, ResNet-18 architecture is adapted with the proper modifications for the speaker recognition task. The modified version will be referred to as modified ResNet-18 (mR18). The comparison between TDNN x-vector model [175], ResNet-18 [69] and mR18 is presented in Figure 3.2. The modifications follow the hypothesis that speaker embedding extractors benefit from features of higher time-resolution. Thus, compared to the original ResNet-18 the maximum pooling at the model beginning has been removed, and the stride in the convolutional blocks has been modified to avoid downsampling along the time dimension, i.e. lowering the time resolution. The last layers of the model have been inspired from the x-vector structure, where statistics pooling and additional segment-level processing with feed-forward layers were employed.

Table 3.1: EER, minDCF results, and approximate network convergence time (in Epoch) for TDNN (x-vector) [175] and mR18 speaker recognition models. Results taken from Publication I.

Network	Softmax	EER [%]	minDCF	Epoch
TDNN	Standard	3,06	0,338	3,10
	AAS [38]	2,57	0,289	7,83
	AdaCos [208]	2,44	0,276	7,53
	ParAda	**2,32**	**0,257**	5,76
mR18	Standard	2,07	0,286	2,82
	AAS [38]	2,12	**0,274**	6,77
	AdaCos [208]	1,94	0,335	5,11
	ParAda	**1,72**	**0,280**	3,51

Table 3.1 presents the summary of the obtained results for the proposed ParAda and mR18 model in reference with to state-of-the-art TDNN model and softmax loss functions, proving the gain offered by the described contributions. The models were trained with combined VoxCeleb1 train part and VoxCeleb2 datasets [123] which were augmented with reverberation and noises. Since the VoxCeleb1 training set was used for training, the VoxCeleb1-E and VoxCeleb1-H trial lists were excluded from tests, which was an accepted practice within the speaker recognition community at the time. ParAda outperformed other existing methods based on softmax losses or their modifications: original, without any modifications (referred in the table as ‘Standard’), angular-based AAS and with adaptive scale (AdaCos). The table also highlights another advantage which is an increased training convergence speed compared to other non-standard softmax-based losses (i.e. AAS and AdaCos). Moreover, a clear improvement can be observed when comparing the results of TDNN (x-vector) [175] and mR18. The mR18 model has also demonstrated its effectiveness in author’s other publications, i.e. for distant speaker verification problem [196] or as a part of the submission in the SdSV Challenge of DSP AGH team [162].

Publication II presents further investigation of the modeling capabilities of the neural network structure for the speaker recognition task. Regarding the mR18 structure, this research seeks methods to extract and preserve speaker information. The hypothesis concerns that the scale-decreased design of the ResNet structure may discard or remove some of the speaker information during processing. The scale-decreased design means that the feature map is downsampled as it is processed by the network. Thus, it was proposed to incorporate SpineNet modeling [42], which represents a scale-permuted design and, moreover, uses multi-scale feature representations before network pooling.



(a) ResNet-50 (b) SpineNet-49

Figure 3.3: Diagram of (a) ResNet-50 and (b) SpineNet-49 residual parts of the structure. Numbers inside blocks denote its level, bold line denote the output blocks, the blocks with solid lines represent the bottleneck type, while dotted represent basic residual.

Scale-permuted means that the processed feature map can decrease or increase during the network processing flow. The terms of scale-decreased and scale-permuted design is presented in Figure 3.3 showing the encoder residual part of ResNet-50 and SpineNet-49. Note that the complete encoder structure, for both models, includes a convolutional layer at the start of the processing pipeline. The numbers inside the blocks represent the level of the block l, which corresponds to the factor with which the feature map is downsampled, which more precisely is 2^{l-2}. Figure 3.3a presents the scale-decreased design, where the processed feature map is steadily downsampled. In case of the ResNet-50 structure, the output from the last block is directly forwarded to the pooling part (not included in Figure 3.3). At the same time, Figure 3.3b presents the SpineNet-49 scale permuted design, where the blocks level have fluctuating order. The blocks with the bold line represent the output blocks, the result of which is forwarded to the pooling part. Here, SpineNet-49 has five output blocks, whose results are properly merged, as they produce feature maps of different sizes. Merging multi-scale feature representations is another key property of this approach, enabling the combination of high-resolution features with low-resolution ones to capture both low-level and high-level information about the utterance.

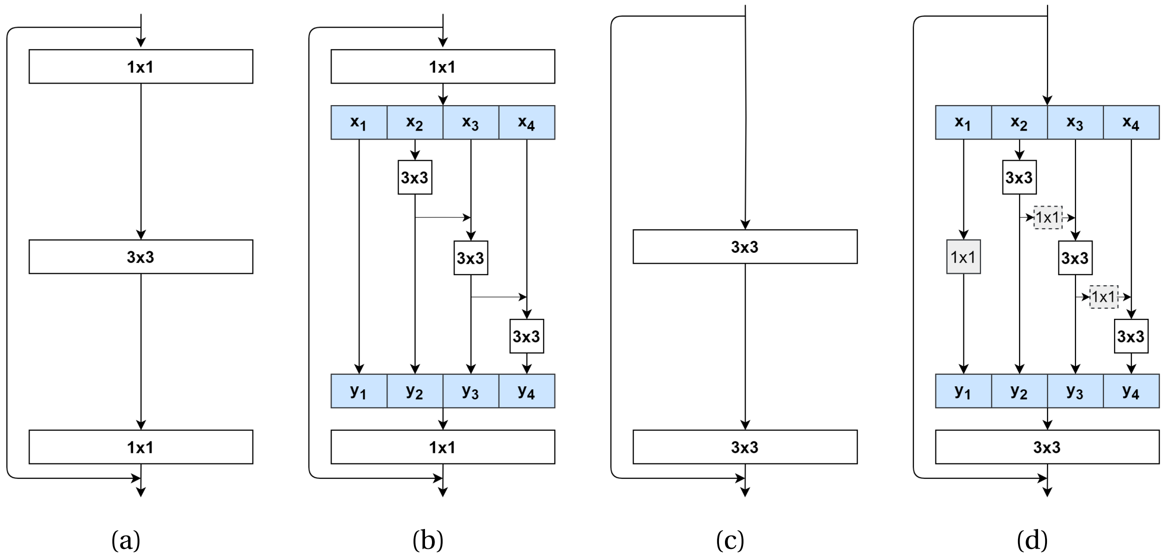


Figure 3.4: Diagrams of (a) ResNet bottleneck block and (b) Res2Net bottleneck block, (c) ResNet basic block and (d) Res2Net basic block. The example presents Res2Net blocks with $s_{\text{Res2Net}} = 4$.

The research presented in Publication II also introduced additional modifications to further increase the receptive field of the model and the level of granularity of processed information through the use of Res2Net blocks [52]. The Res2Net structure was originally introduced for the image processing tasks. It is also a method of processing the multi-scale features within bottleneck residual block. The multi-scale processing, i.e. processing the information in its small details as well as the broad general picture allows to obtain much more accurate information. To illustrate this concept more clearly, let us consider examples from image processing. Firstly, multi-scale processing allows to take into account the variable sizes of the information provided at the input, e.g. in image processing the goal is to detect the big wardrobe as well as a small cup. Secondly, the context of the information may be much bigger than the information itself, e.g. having the knowledge and context of the picture of a kitchen helps to decide whether a small red object is an apple or ball. Lastly, the multi-scale understanding of the picture allows to capture both details as well as the meaning of the entire image, e.g. when identifying a small gray patch in an image, it is hard to distinguish whether it is a cat fur or fragment of a concrete wall, but looking at the bigger picture and detecting eyes, nose, and whiskers allows to take a correct decision.

The same principle can be applied to speech. Short speech segments may provide only limited cues, but multi-scale analysis allows combining fine-grained acoustic details (e.g. pitch, timbre) with broader patterns (e.g. rhythm, prosody, speaking style).

To achieve the mentioned goals, Res2Net introduced proper architecture modifications to the bottleneck residual block. The diagrams of the bottleneck residual block and the corresponding Res2Net bottleneck block are presented, respectively, in Figures 3.4a and 3.4b. The output of the first 1 × 1 convolutional layer is divided into s_{Res2Net} chunks, denoted as x, where each of them has the same feature map size, but $\frac{1}{s_{\text{Res2Net}}}$ channels. The middle layer (3 × 3 convolution with n channels) is replaced by a collection of 3 × 3 convolutional layers with w channels, where $n = w \times s_{\text{Res2Net}}$. Each input chunk is processed by its own 3 × 3 convolutional layer (except x_1). The new layers are connected in the hierarchical manner, where each consecutive block presents an increased scale of the features. With each pass through the 3 × 3 convolution, the receptive field expands, which leads to multiple feature scales as a result of the combination effect. This way Res2Net introduces s_{Res2Net} - a *scale* dimension. Original work [52] introduced Res2Net blocks for bottleneck type. As SpineNet structure is built of both bottleneck and basic residual blocks, in the Publication II the Res2Net version for the basic block has been introduced. The comparative diagram in Figure 3.4 presents the ResNet bottleneck and basic blocks, as well as the corresponding Res2Net bottleneck and basic blocks. In contrast to the bottleneck block, the basic Res2Net may include optional 1 × 1 convolution for x_1 chunk and 1 × 1 convolution projections between inner residual connections. Additional projections are required when the w channels are increased with respect to $\frac{C_{in}}{s_{\text{Res2Net}}}$ (C_{in} - input number of channels), which may be applied when there is a need to preserve or extend network complexity or to improve performance.

New model structure included Squeeze-and-Excitation (SE) blocks [76] modified for for the speaker recognition task, namely the Time-Squeeze-and-Excitation (T-SE) module [87, 163]. The goal of SE module is to recalibrate the dependencies between channels by capturing and modeling their interdependencies. Its processing flow is based on two steps: Squeeze and Excitation. The first step aggregates information with a pooling applied on the entire feature maps. Next, the Excitation step follows with a self-gating mechanism, which is based on two feed-forward layers, with a reduction factor r to force learning of the compact and general representations. The difference between the modules is that originally the SE block was proposed to recalibrate the dependencies between channels only, while the T-SE block extends the recalibration to the channel and frequency

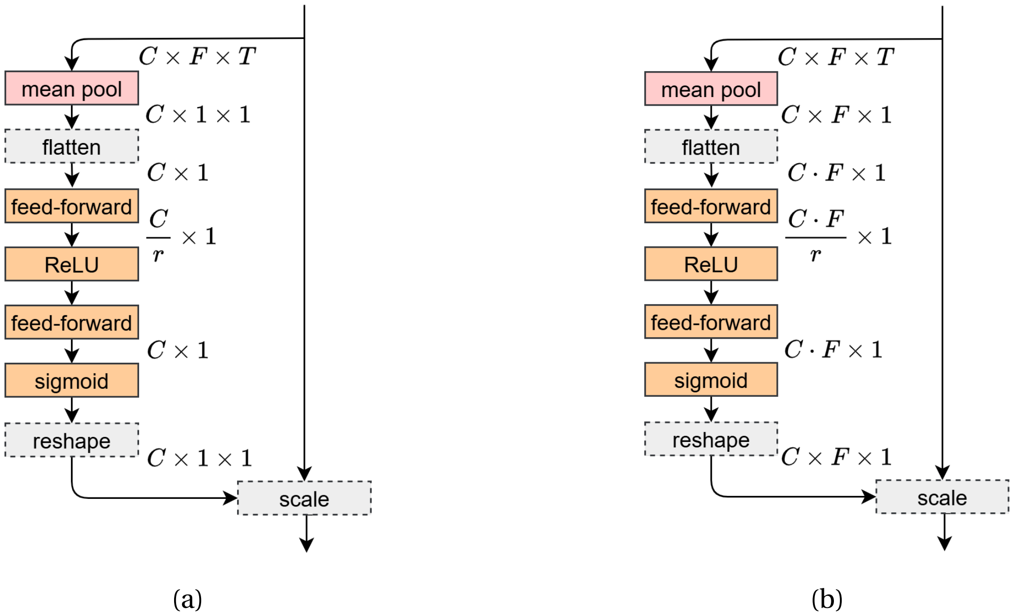


Figure 3.5: The schemes of the (a) Squeeze-Excitation and (b) Time-Squeeze-Excitation blocks. C stands for channel dimension, F - feature dimension, T - time dimension, r - reduction factor.

domain. The comparative scheme of the processing flow of the SE and T-SE blocks is presented in Figure 3.5.

For this research, the evaluation was performed on the VoxCeleb dataset. VoxCeleb2 was used for training, while VoxCeleb1 was used as a test set with trials: Extended (VoxCeleb1-E), Hard (VoxCeleb1-H) and Original (VoxCeleb1-O). Table 3.2 presents an excerpt of the results showed in Publication II. The minDCF is presented with $P_{tar} = 0.05$ (DCF5) and $P_{tar} = 0.01$ (DCF1). The table also includes the number of trainable parameters for each network, as well as the FLOP (Floating Point Operations) value, which represents the number of multiply-add operations, counted as a single operation, calculated for a 3-second audio fragment. The T-SE-Res2Net-50 and T-SE-Spine2Net-49 represent ResNet-50 and SpineNet-49 with introduced final modifications of the Res2Net and T-SE modules. As can be observed, the presented model, along with added modifications, provides prime performance. Moreover, despite an increase in parameter number, it is characterized by a much lower FLOP number, which shows increased efficiency given the model size.

Table 3.2: Excerpt of the results from Publication II for ResNet and SpineNet based structures for VoxCeleb1 test dataset.

			VoxCeleb1-E			VoxCeleb1-H			VoxCeleb1-O		
Network	# Params	# FLOPs	EER	DCF5	DCF1	EER	DCF5	DCF1	EER	DCF5	DCF1
ResNet-34	25.5M	27.3G	1.19	0.078	0.140	2.27	0.137	0.219	1.35	0.088	0.146
ResNet-50	35.6M	30.7G	1.30	0.082	0.150	2.33	0.142	0.235	1.44	0.100	0.173
SpineNet-49	28.6M	26.0G	1.17	0.074	0.129	2.14	0.129	0.213	1.11	0.088	0.125
T-SE-Res2Net-50	88.1M	32.1G	1.05	0.067	0.117	1.95	0.113	0.196	1.12	0.071	0.103
T-SE-Spine2Net-49	58.0M	26.2G	0.99	0.065	0.112	1.95	0.117	0.192	0.92	0.068	0.105

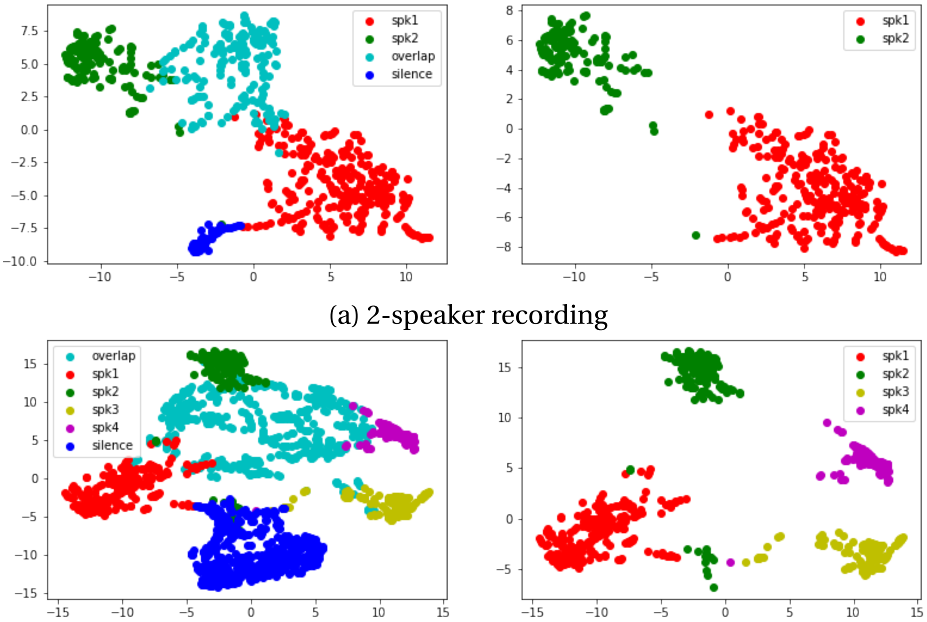
3.2 Non-Autoregressive Attractor estimation

The following section describes the contributions in the areas of speaker diarization and joint speaker diarization and separation tasks. The main proposal is the method called Non-Autoregressive Attractor (NAA) estimation. NAA was introduced and developed by the author of this thesis within the speaker diarization task, and later applied for the joint speaker diarization and separation task.

3.2.1 Speaker diarization

The next part of the research work was the problem of proper extraction of speaker information for the speaker diarization task. As described in detail in Chapter 2, the most popular method of speaker diarization consists in clustering of speaker embeddings extracted from segmented audio. Thus, speaker diarization is a natural extension of the speaker recognition task. End-to-end neural speaker diarization (EEND) models are the next generation of state-of-the-art approaches for speaker diarization. Frame-level embeddings have been shown to form speaker and silence clusters, with intermediate overlap embeddings [73, 72], which also can be observed in examples presented in Figure 3.6, adopted from Publication IV. EEND-EDA introduces an attractor mechanism into the EEND model to deal with unknown number of speakers. However, the LSTM-based autoregressive attractor generator makes the process obscure due to LSTM nature, i.e. sequential dependence and nonlinear processing. In order to ensure that attractor decision making is explainable, the Non-Autoregressive Attractor (NAA) estimator was proposed in Publication III.

The NAA idea operates on the property of the clusters formed by frame-level embeddings. The diagram of the EEND with NAA, shortly - EEND-NAA, is depicted in Figure 3.7. Frame-level embeddings from the EEND encoder are used as input for the k-means clustering. The number of clusters is set to be equal to the number of speakers. The clustering



(a) 2-speaker recording

(b) 4-speaker recording

Figure 3.6: T-SNE visualization of the encoder embeddings obtained by EEND-NAA (with $I = 4$). Left plots: all frame-level embeddings. Right plots: embeddings containing only a single speaker. Plots adopted from Publication IV.

result is refined with Transformer decoder layers, where the attention mechanism helps further in producing discriminative final attractors. The k-means cluster centers are used as an initialization for the final attractor representations. However, the clustering is used directly on top of all frame-level embeddings, which means that the cluster centers are computed not only with single-speaker embeddings but also silence and overlap ones. Thus, the initialization may be contaminated with ambiguous information. In order to overcome this challenge, an iterative refinement step was proposed in Publication III. After obtaining the diarization result using k-means initialization, the result is reused to re-estimate the initial attractors, which constitutes the new cluster assignment. Using this approach, cluster centers can be refined and computed only from embeddings that belong to a particular speaker. For some of the recordings, iterative refinement had a notable impact which is presented in Figure 3.8, which depicts the diarization system decisions for each iteration step. In the beginning the model classifies most of the embeddings as silence or overlap. With the next iterations we can observe that system refines its decisions, leading to correct assignment of the diarization decision.

By definition, the attractors are supposed to encode the within-recording relative speaker information. In order to further enhance the attractor discriminability, the pro-

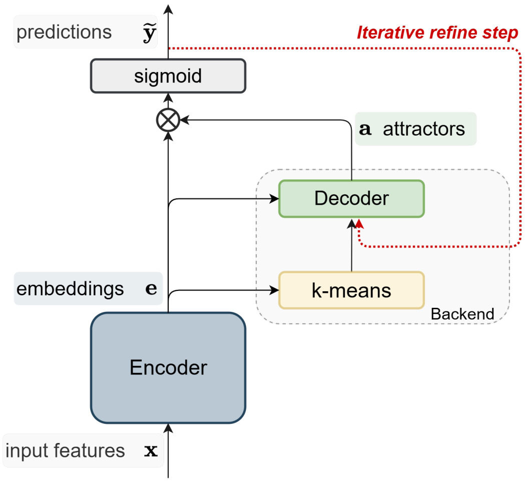
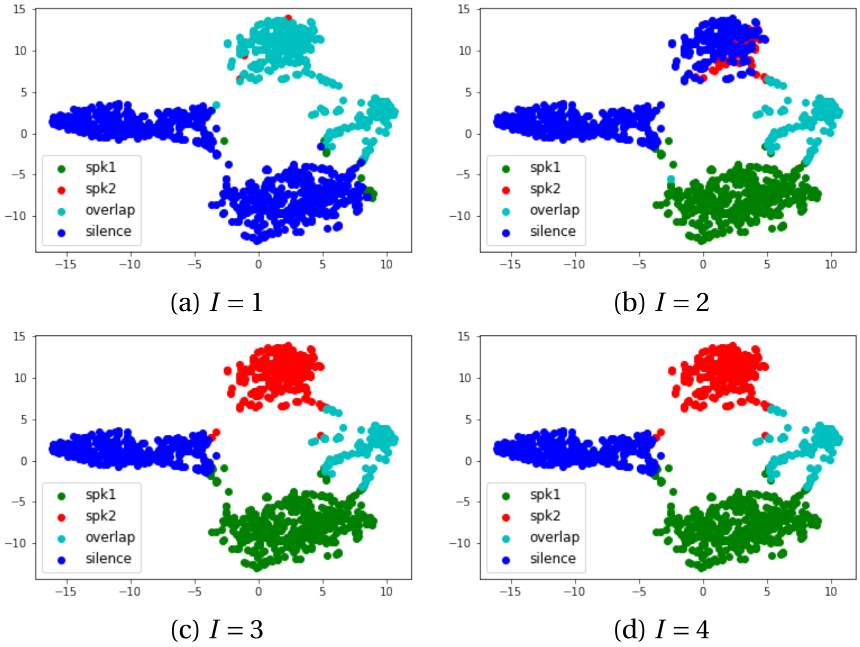


Figure 3.7: The scheme of the EEND with Non-Autoregressive Attractor estimation proposed in Publication III.



(a) $I = 1$

(b) $I = 2$

(c) $I = 3$

(d) $I = 4$

Figure 3.8: EEND-NAA encoder embedding visualization at each I-th refinement step. Decisions obtained from the system output serve as labels. Plots adopted from Publication III.

posed model incorporates an additional speaker classification loss applied to the final attractor representations. The excerpt of the publication results is presented in Table 3.3 on the CALLHOME (CH) dataset [147]. The presented table demonstrates the improvement of the proposed method over the baseline, showing a consistent decrease in DER values as the number of iterations increases. The final performance corresponds to the model incorporating the speaker classification loss in addition to the diarization loss.

Table 3.3: Diarization Error Rate (DER) performance for the EEND-EDA and EEND-NAA with different number of iterative refinement steps. '+ $\mathscr{L}_{\text{spk}}$' indicates model with included speaker classification loss.

Model	CH
EEND-EDA	9.24
EEND-NAA, $I = 1$	8.94
EEND-NAA, $I = 2$	8.19
EEND-NAA, $I = 3$	8.10
EEND-NAA, $I = 4$	7.94
EEND-NAA, $I = 4 + \mathscr{L}_{\text{spk}}$	**7.83**

Following this proposal (Publications III and IV), several non-autoregressive solutions began to appear in the literature [49, 26]. Nevertheless, the described research was the first to propose the Non-Autoregressive Attractor estimation for the speaker diarization task. The initial proposal (Publication III) had certain limitations, as the model required prior knowledge of the number of speakers. Moreover, the presented evaluation was conducted only for recordings containing two speakers. For these reasons Publication IV introduced three possible extensions of the originally proposed solution to the variable and unknown speaker conditions.

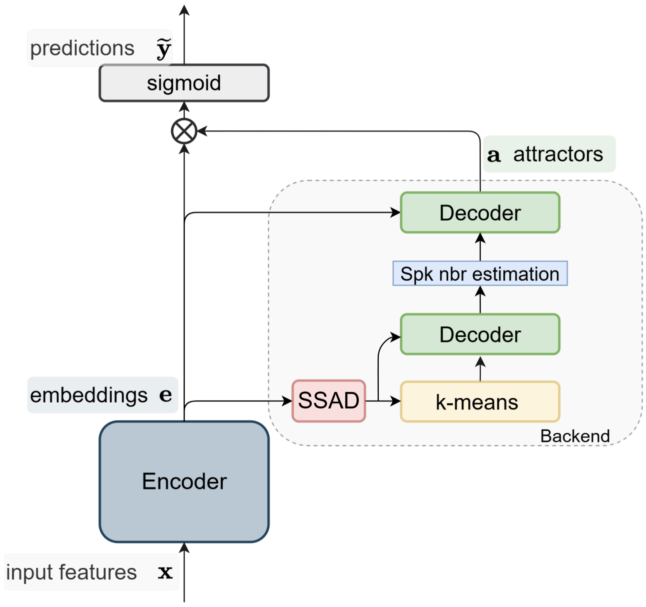


Figure 3.9: The processing flow of the EEND-NAA-Overest system.

All the extensions share three similar modifications: introduction of the Single Speaker Activity Detection (SSAD) module, modification of the goal of the clustering step, and adding an additional Transformer decoder module. The first extension, the so-called EEND-NAA-Overest, is presented in Figure 3.9. Its processing flow is as follows: first, the frame-level embeddings from the encoder are processed by the SSAD module. The task of the SSAD module is to filter out embeddings containing more than one speaker (overlap) or none (silence). This way, the k-means clustering operates only on the embeddings that come from one speaker at a time. Filtered representations serve as input for the k-means clustering algorithm. However, contrary to the initial proposal in Publication III, in this method, the clustering algorithm does not cluster with a number equal to the number of speakers. Instead, it uses an arbitrarily chosen value that overestimates the speaker count, exceeding the maximum expected in the recording. By doing so, the diarization can be performed for the unknown speaker number. Next, the obtained representations are fed to the first decoder, whose task is to refine the centers to single-speaker representations, by pulling true speaker centers closer while pushing irrelevant ones farther away, enabling the subsequent linear layer to distinguish one center per speaker. In order to facilitate that, the proper loss is applied on top of the representations after the first decoder, which is based on the softmax cross-entropy loss. The centers selected as speaker ones are compared to ideal speaker representations. In this work, ideal speaker representations are obtained by computing the mean embedding across all single-speaker frame-level embeddings in which the respective speaker is present. The assignment of cluster centers to speakers is done with a permutation approach. Given K cluster centers and S ideal speaker centers, all possible S-sized subsets of the K are considered. For each subset and its permutations, the distances to the ideal centers are computed, and the permutation with the lowest total distance determines which centers are selected as speaker ones. The refined centers are then processed by a simple linear layer whose task is to decide which attractors belong to a speaker or not, which also represents a step where speaker number is counted. After selecting the speaker representations, the chosen pre-attractor representations are forwarded to the second decoder, which produces the final attractors.

Figure 3.10 presents the second system, which is referred to as EEND-NAA-2step. It has similar processing steps as EEND-NAA-Overest, but the NAA-2step module performs a slightly different task. Similarly, first, the encoder embeddings are processed by the SSAD module. Also, on the top of the filtered embeddings, the k-means with an overestimated number of clusters is applied. However, the goal of the first decoder is different. The EEND-NAA-2step follows the assumption that all cluster centers represent speakers, which

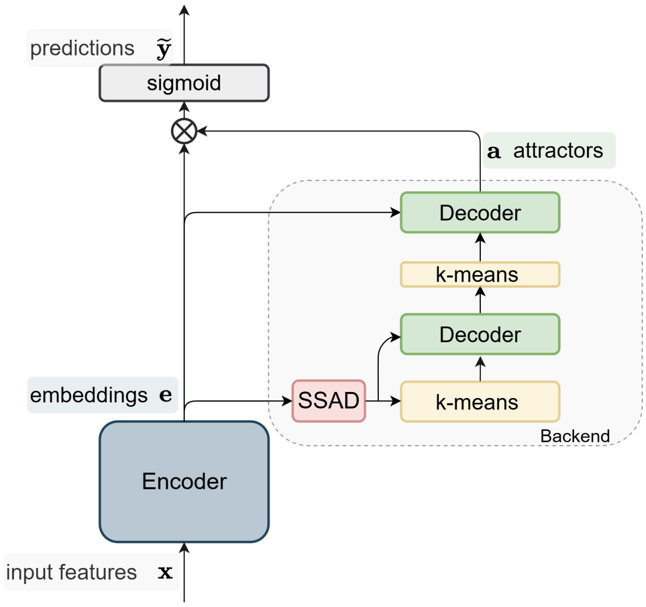


Figure 3.10: The processing flow of the EEND-NAA-2step system.

means that one speaker can be represented by more than one cluster center. Thus, the task of the first decoder is to refine the centers in such a way that the centers that come from the same speaker are brought closer, and the ones that are from different speakers are pushed away from each other. The property is achieved by applying a contrastive loss. This allows the centers to be merged with the second clustering that is present after the first decoder. During inference, the second clustering is replaced by spectral clustering. The eigenvalue analysis is applied to count the speaker number. During training, in order to save computation time, k-means clustering is used. Finally, the merged representations are processed by the second decoder that outputs the final attractor representations.

Figure 3.11 presents the third and last extension of the proposed systems. It is referred to as EEND-NAA-1step and represents a simplified EEND-NAA-2step model. Similarly as in the previous systems, encoder embeddings are processed by the SSAD module and fed to the clustering algorithm. Contrary to previous approaches, the clustering step sets the number of centers equal to the number of speakers. Thus, the speaker counting step is performed at this stage of the backend processing, immediately after SSAD filtering. During inference, spectral clustering is incorporated to estimate the number of speakers. It is replaced with k-means for the training time, in order to save on computations. Then, the obtained centers are processed by the first decoder, and, similarly to EEND-NAA-Overest, the loss based on the softmax cross-entropy is applied to refine the centers to

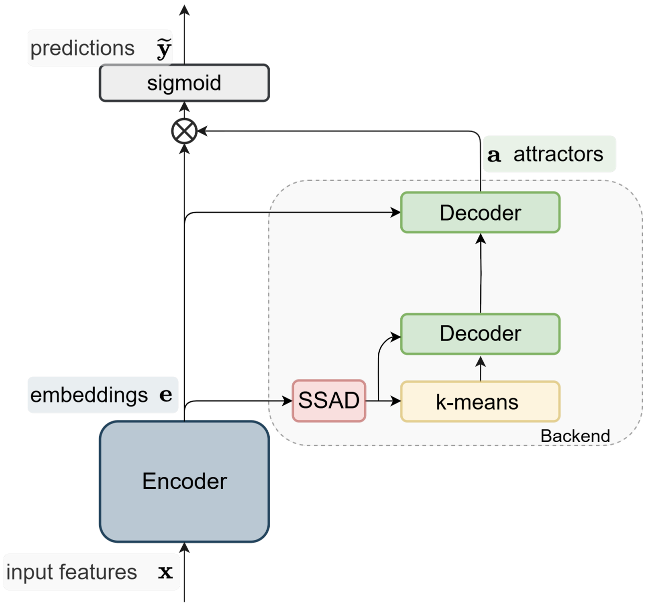


Figure 3.11: The processing flow of the EEND-NAA-1step system.

ideal speaker representations. Next, the second decoder processes the embeddings, and the final attractors are retrieved.

Table 3.4: DER results for simulated test recordings for estimated number of speakers. Test sets results includes sets of recordings with 1, 2, 3 and 4 speakers, and average (Avg) among all test recordings.

	#Speakers				
System	1	2	3	4	Avg
EEND-EDA	0.46	6.07	12.30	15.75	8.65
EEND-NAA-Overest	0.15	4.25	9.12	12.84	8.28
EEND-NAA-2step	0.08	4.31	6.83	12.26	5.87
EEND-NAA-1step	1.01	5.08	8.76	14.78	7.40

Table 3.4 shows fragment of the DER results from Publication IV. It presents the performance of the proposed systems, compared to EEND-EDA baseline (in Publication IV this model is referred as ‘EEND-EDA, PyTorch’), for models trained and evaluated on simulated recordings. In these results, the models estimated the speaker number (i.e. the speaker number is unknown for the systems). The test results include sets of recordings with 1, 2, 3 and 4 speakers, as well as average among all test recordings, which clearly demonstrate the advantage of the proposed methods over the baseline.

3.2.2 Joint speaker diarization and separation

Speaker separation and speaker diarization are closely related tasks. Both aim to track speakers, but at different granularity levels. Diarization returns speaker activity, a binary decision whether a particular speaker is present in the particular time frame. Speaker separation outputs a more detailed information in the form of speech waveform, which is typically obtained through masks that represent speaker activity but in the time and frequency domain. For this reason, recently, attention has been directed towards combining these two tasks into one system in order to cope with their challenges and problems. Publication V presents research in which the effective method of non-autoregressive attractor estimation has been combined into a separation network to perform a joint separation and diarization task and bridge the gap between speaker separation and diarization by merging these related tasks into unified framework.

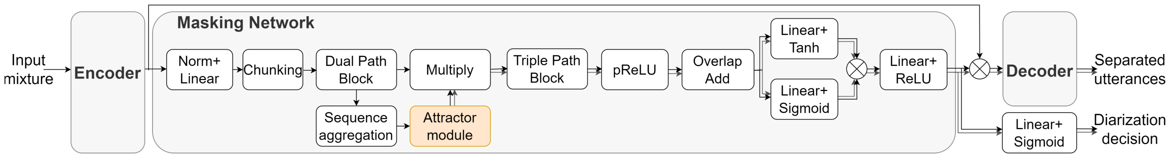


Figure 3.12: Scheme of the SepDiar architecture. Diagram adopted from Publication V.

The proposal of the joint speaker diarization and separation architecture builds on the SepFormer-based separation structure - SepEDA. The separation model is composed of three main building blocks: encoder, masking network, and decoder. The model's input is a raw audio mixture, and output is the separated audio tracks for each speaker. The encoder task is to transform the time representation of the recording into the time-feature domain. Next, the masking network produces speaker masks that multiplied with encoder features serve as decoder input. Finally, the decoder reconstructs masked representations into time representations. The architecture employs attractor generation in the middle of the masking network, which allows for extraction of the speaker representations and processing recordings with flexible number of speakers. Proposed model extends separation with a diarization part and loss objective function in order to perform the joint task. It is referred to as SepDiar, and its general diagram is presented in Figure 3.12. The proposal also includes two attractor generation variants for different conditions: conversational with low speech overlap, and for very high speech overlap, characteristic for speaker separation benchmarking.

The first one, Cluster-based Attractor (CA) mechanism is directly inspired from the diarization work described in Publications III and IV. The diagram of the attractor generation mechanism is shown in Figure 3.13. It borrows the basic idea of filtering embeddings with the SSAD module and applies the k-means clustering on top of it, leveraging the speaker information from the single-speaker regions.

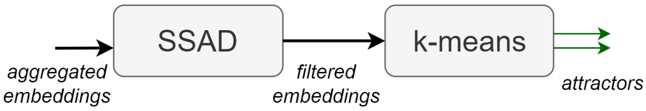


Figure 3.13: Cluster-based Attractor (CA) scheme. Adopted from Publication V.

However, the CA mechanism, which is dedicated to scenarios with time periods with activity of a single speaker, should not be expected to demonstrate its effectiveness in typical for separation high-overlap conditions with recordings that contain very little or no single-speaker regions. The Diarization-based Attractors (DA) generation was designed to address that problem. Its scheme is depicted in Figure 3.14. It follows the processing flow that is similar to the EEND-VC processing [95]. The aggregated embeddings at the previous processing step are processed by DA encoder, whose output is processed in parallel by two blocks: speaker and diarization. The task of the diarization block is to estimate the diarization result. The second block - speaker - processes the input into separate embedding sequences, where the number of sequences is equal to the assumed maximum number of speakers. The final attractors are obtained as a weighted average of the speaker embeddings with diarization speaker activity results as weights.

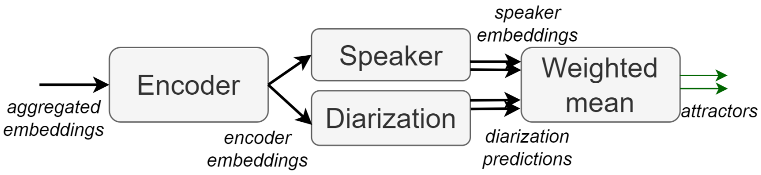


Figure 3.14: Diarization-based Attractor (DA) scheme. Adopted from Publication V.

In the proposed SepDiar framework, the final diarization is obtained by applying a single linear layer with a Sigmoid activation at each time bin of speaker separation masks (also presented in Figure 3.12). The results presented in Publication V reveal the top performance of the proposed methods, including typical simulated separation datasets, as well as real-life recordings used to benchmark diarization-only models, and presents the versatility of the NAA method for diarization and separation tasks. The fragment of these results is presented in Table 3.5, where selected results are presented for SparseLibri2Mix with 40% speech overlap and diarization-only CALLHOME dataset. EEND-SS [111] and EEND-EDA [111] are results from [111] paper, while EEND-EDA (Ours) is an EEND-EDA model trained independently in the course of this research, SepFormer and SepEDA are separation baselines, and SepDiarCA (SepDiar with CA) and SepDiarDA (SepDiar with DA) are the proposed models. The results for SparseLibri2Mix present performance for both separation and diarization tasks. The proposed systems consistently outperform all baseline systems, both for joint and individual tasks. The further evaluation on CALLHOME (CH) dataset demonstrate the effectiveness on real-life recordings for diarization task. As can be observed, the obtained results confirm the effectiveness of the proposed approaches.

Table 3.5: SparseLibri2Mix (test set with 40% speech overlap) and CALLHOME (CH) results. * indicates the results derived from the plot diagram in [111].

	SparseLibri2Mix		CH
System	SI-SDRi	DER	DER
EEND-SS [111]*	7.5	5.4	–
EEND-EDA [111]*	–	10.3	–
EEND-EDA (Ours)	–	9.99	21.47
SepFormer	18.72	–	–
SepEDA	17.45	–	–
SepDiarCA	19.14	2.14	**6.80**
SepDiarDA	**19.67**	**2.00**	7.40

Chapter 4

Finale

4.1 Summary

Since first successful application of deep neural networks, the performance and application of many advances in speech technology have improved drastically. Many systems are able to show performance sometimes on par or better than human accuracy. The developed approaches often strive for better accuracy and generalization by seeking optimal solutions and leveraging knowledge and advances from other speech processing tasks.

This thesis confirms the hypotheses stated in Chapter 1.2 and presents substantial contributions to the fields of speaker recognition and diarization aiming at optimization of DNN-based approaches. More specifically, the research concerns various aspects of speaker representations for these tasks.

Publications I and II introduce several optimizations of the neural structure towards more discriminative speaker representations for the speaker recognition task. They introduce an angular-based objective function, which adapts its parameters based on the current convergence step and accuracy in order to provide optimal training, and as a result provides an improved performance and convergence speed of the training. In both Publications, the presented modifications to the structure are aimed to increase the focus on temporal and frequency dependencies by combining multi-scale features, which leads to better discrimination and information about speakers.

Publications III and IV propose a generic non-autoregressive attractor generation for estimation of speaker representations for the diarization task, which presents an improved performance over baseline methods and directly exploits information provided by the encoder of the diarization network. Namely, the method incorporates speaker information encoded in the frame-level embeddings, which allows to extract and refine speaker representations from the recording in an explainable manner. The method has been investigated and developed into different variants to address generic conditions of an unknown and flexible number of speakers. The evaluation was conducted on simulated recordings as well as real-life test sets such as CALLHOME and DIHARD.

Publication V continues to exploit non-autoregressive approach for joint speaker diarization and separation tasks, proving the versatility of the proposed method. Moreover, the research proposes a framework that merges the tasks of separation and diarization. The results show improved accuracy of the two proposed methods for high-overlap conditions, typical for the separation task, and sparse-overlap conditions, typical for conversations and the diarization task.

4.2 Future work

There are multiple directions that the presented research can take. First, further research on speaker information encoded in attractors would be beneficial. This study could examine whether diarization attractors can also be exploited in the context of the speaker recognition task. The investigation would be valuable in order to increase the versatility of the framework towards the joint speaker recognition and diarization task. Moreover, if the generated speaker attractors represent the absolute speaker characteristics, the idea could set a research direction into extraction of the overlap-robust speaker embeddings from multi-speaker recordings. The next goal is to extend the properties of the diarization and joint diarization and separation models to deal with long recordings. Currently, the joint model presents good performance on relatively short recordings (up to 60 seconds), which is a scenario not fully applicable for some speaker diarization use cases (aiming to process recordings from minutes to even hours long) or continuous speech separation. This problem can be addressed by performing separation and diarization on short chunks, extracting the attractors, and then combining them into a single stream per speaker based on attractor similarity across chunks. Moreover, that research direction could evolve into exploration of an online/streaming diarization and separation task. Further development could aim to integrate the proposed solutions with the ASR, both as a downstream task evaluation, as well as integrating the ASR model into a joint task in order to fully answer the question "who spoke when and what". Finally, one of the problems in evaluation of the joint of diarization and separation framework is the lack of real-life ground-truth examples for the separation task, which is required in order to train the network in a supervised manner. In order to overcome the problem, the combination of unsupervised techniques for separation and supervised for diarization could allow processing and leveraging recordings from different datasets, increasing the available amount of training data.

Bibliography

[1] Abdul, Z. K. and Al-Talabani, A. K. (2022). Mel Frequency Cepstral Coefficient and its Applications: A Review. *IEEE Access*, 10:122136–122158.

[2] Anguera, X., Wooters, C., Peskin, B., and Aguiló, M. (2006). Robust Speaker Segmentation for Meetings: The ICSI-SRI Spring 2005 Diarization System. In Renals, S. and Bengio, S., editors, *Machine Learning for Multimodal Interaction*, pages 402–414, Berlin, Heidelberg. Springer Berlin Heidelberg.

[3] Aronowitz, H. and Aronowitz, V. (2010). Efficient score normalization for speaker recognition. In *ICASSP 2010 - 2010 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4402–4405.

[4] Aronowitz, H., Irony, D., and Burshtein, D. (2005). Modeling intra-speaker variability for speaker recognition. In *Interspeech 2005*, pages 2177–2180.

[5] Aronowitz, H. and Zhu, W. (2020). Context and Uncertainty Modeling for Online Speaker Change Detection. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8379–8383.

[6] Arora, A., Raj, D., Subramanian, A. S., Li, K., Ben-Yair, B., Maciejewski, M., Zelasko, P., Garcia, P., Watanabe, S., and Khudanpur, S. (2020). The JHU Multi-Microphone Multi-Speaker ASR System for the CHiME-6 Challenge. In *6th International Workshop on Speech Processing in Everyday Environments (CHiME 2020)*, pages 48–54.

[7] Atal, B. (1976). Automatic recognition of speakers from their voices. *Proceedings of the IEEE*, 64(4):460–475.

[8] Auckenthaler, R., Carey, M., and Lloyd-Thomas, H. (2000). Score Normalization for Text-Independent Speaker Verification Systems. *Digital Signal Processing*, 10:42–54.

[9] Bai, Z. and Zhang, X.-L. (2021). Speaker recognition based on deep learning: An overview. *Neural Networks*, 140:65–99.

[10] Bishop, C. (2006). *Pattern Recognition and Machine Learning*, volume 16, pages 140–155. Springer.

[11] Boeddeker, C., Subramanian, A. S., Wichern, G., Haeb-Umbach, R., and Le Roux, J. (2024). TS-SEP: Joint Diarization and Separation Conditioned on Estimated Speaker Embeddings. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:1185–1197.

[12] Bozonnet, S., Evans, N., Anguera, X., Vinyals, O., Friedland, G., and Fredouille, C. (2010). System output combination for improved speaker diarization. In *Interspeech 2010*, pages 2642–2645.

[13] Bredin, H. (2017). TristouNet: Triplet loss for speaker turn embedding. In *ICASSP 2017 - 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5430–5434.

[14] Brümmer, N. and du Preez, J. (2006). Application-independent evaluation of speaker detection. *Computer Speech & Language*, 20(2):230–275. Odyssey 2004: The speaker and Language Recognition Workshop.

[15] Brümmer, N., van Leeuwen, D., and Swart, A. (2014). A comparison of linear and non-linear calibrations for speaker recognition. In *The Speaker and Language Recognition Workshop (Odyssey 2014)*, pages 14–18.

[16] Bullock, L., Bredin, H., and García-Perera, L. P. (2019). Overlap-Aware Diarization: Resegmentation Using Neural End-to-End Overlapped Speech Detection. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7114–7118.

[17] Burget, L., Matejka, P., Schwarz, P., Glembek, O., and Cernocky, J. H. (2007). Analysis of Feature Extraction and Channel Compensation in a GMM Speaker Recognition System. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(7):1979–1986.

[18] Cai, D., Qin, X., Cai, W., and Li, M. (2019). The DKU System for the Speaker Recognition Task of the 2019 VOiCES from a Distance Challenge. In *Interspeech 2019*, pages 2493–2497.

[19] Cai, W., Chen, J., and Li, M. (2018). Exploring the Encoding Layer and Loss Function in End-to-End Speaker and Language Recognition System. In *The Speaker and Language Recognition Workshop (Odyssey 2018)*. ISCA.

[20] Campbell, W., Campbell, J., Reynolds, D., Singer, E., and Torres-Carrasquillo, P. (2006a). Support vector machines for speaker and language recognition. *Computer Speech & Language*, 20(2):210–229. Odyssey 2004: The speaker and Language Recognition Workshop.

[21] Campbell, W., Sturim, D., and Reynolds, D. (2006b). Support vector machines using GMM supervectors for speaker verification. *IEEE Signal Processing Letters*, 13(5):308–311.

[22] Campbell, W., Sturim, D., Reynolds, D., and Solomonoff, A. (2006c). SVM Based Speaker Verification using a GMM Supervector Kernel and NAP Variability Compensation. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2006*, volume 1, pages I – I.

[23] Carletta, J., Ashby, S., Bourban, S., Flynn, M., Guillemot, M., Hain, T., Kadlec, J., Karaiskos, V., Kraaij, W., Kronenthal, M., et al. (2005). The AMI meeting corpus: A pre-announcement. In *International workshop on machine learning for multimodal interaction*, pages 28–39. Springer.

[24] Chen, S., Gopalakrishnan, P. S., and Watson, I. T. J. (1998). Speaker, Environment and Channel Change Detection and Clustering via the Bayesian Information Criterion. In *Proc. DARPA Broadcast news transcription and understanding workshop*, volume 8, page 127–132.

[25] Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S., Chen, Z., Li, J., Kanda, N., Yoshioka, T., Xiao, X., Wu, J., Zhou, L., Ren, S., Qian, Y., Qian, Y., Wu, J., Zeng, M., Yu, X., and Wei, F. (2022). WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.

[26] Chen, Z., Han, B., Wang, S., and Qian, Y. (2023). Attention-based Encoder-Decoder Network for End-to-End Neural Speaker Diarization with Target Speaker Attractor. In *Proc. Interspeech 2023*, pages 3552–3556.

[27] Chen, Z., Ren, Z., and Xu, S. (2019). A Study on Angular Based Embedding Learning for Text-independent Speaker Verification. *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*.

[28] Chen, Z., Yoshioka, T., Lu, L., Zhou, T., Meng, Z., Luo, Y., Wu, J., Xiao, X., and Li, J. (2020). Continuous Speech Separation: Dataset and Analysis. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7284–7288.

[29] Cho, Y. D. and Kondoz, A. (2001). Analysis and improvement of a statistical model-based voice activity detector. *IEEE Signal Processing Letters*, 8(10):276–278.

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

[31] Cortes, C. and Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, 20(3):273–297.

[32] Cosentino, J., Pariente, M., Cornell, S., Deleforge, A., and Vincent, E. (2020). Librimix: An open-source dataset for generalizable speech separation. *arXiv preprint arXiv:2005.11262*.

[33] Davis, A., Nordholm, S., and Togneri, R. (2006). Statistical Voice Activity Detection Using Low-Variance Spectrum Estimation and an Adaptive Threshold. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(2):412–424.

[34] Davis, S. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366.

[35] Dehak, N., Dehak, R., Kenny, P., Brümmer, N., Ouellet, P., and Dumouchel, P. (2009). Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification. In *Interspeech 2009*, pages 1559–1562.

[36] Dehak, N., Kenny, P. J., Dehak, R., Dumouchel, P., and Ouellet, P. (2011). Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4).

[37] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

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

[39] Desplanques, B., Thienpondt, J., and Demuynck, K. (2020). ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification. In *Interspeech 2020*, pages 3830–3834.

[40] Diez, M., Burget, L., and Matejka, P. (2018). Speaker Diarization based on Bayesian HMM with Eigenvoice Priors. In *The Speaker and Language Recognition Workshop (Odyssey 2018)*, pages 147–154.

[41] Diez, M., Burget, L., Wang, S., Rohdin, J., and Černocký, J. (2019). Bayesian HMM Based x-Vector Clustering for Speaker Diarization. In *Proc. Interspeech 2019*, pages 346–350.

[42] Du, X., Lin, T. Y., Jin, P., Ghiasi, G., Tan, M., Cui, Y., Le, Q. V., and Song, X. (2020). SpineNet: Learning Scale-Permuted Backbone for Recognition and Localization. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11589–11598.

[43] Enqing, D., Guizhong, L., Yatong, Z., and Xiaodi, Z. (2002). Applying support vector machines to voice activity detection. In *6th International Conference on Signal Processing, 2002.*, volume 2, pages 1124–1127 vol.2.

[44] Fang, X., Ling, Z.-H., Sun, L., Niu, S.-T., Du, J., Liu, C., and Sheng, Z.-C. (2021). A Deep Analysis of Speech Separation Guided Diarization Under Realistic Conditions. In *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 667–671.

[45] Ferrer, L. and Mclaren, M. (2020). A Speaker Verification Backend for Improved Calibration Performance across Varying Conditions. In *The Speaker and Language Recognition Workshop (Odyssey 2020)*, pages 372–379.

[46] Fiscus, J., Ajot, J., Michel, M., and Garofolo, J. (2006). The Rich Transcription 2006 Spring Meeting Recognition Evaluation. pages 309–322.

[47] Fujita, Y., Kanda, N., Horiguchi, S., Nagamatsu, K., and Watanabe, S. (2019a). End-to-End Neural Speaker Diarization with Permutation-Free Objectives. In *Proc. Interspeech 2019*, pages 4300–4304.

[48] Fujita, Y., Kanda, N., Horiguchi, S., Xue, Y., Nagamatsu, K., and Watanabe, S. (2019b). End-to-End Neural Speaker Diarization with Self-attention. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 296–303.

[49] Fujita, Y., Komatsu, T., Scheibler, R., Kida, Y., and Ogawa, T. (2023). Neural Diarization with Non-Autoregressive Intermediate Attractors. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

[50] Fujita, Y., Watanabe, S., Horiguchi, S., Xue, Y., Shi, J., and Nagamatsu, K. (2020). Neural Speaker Diarization with Speaker-Wise Chain Rule. In *arXiv preprint arXiv:2006.01796*.

[51] Fukuda, T., Ichikawa, O., and Nishimura, M. (2010). Long-Term Spectro-Temporal and Static Harmonic Features for Voice Activity Detection. *IEEE Journal of Selected Topics in Signal Processing*, 4(5):834–844.

[52] Gao, S.-H., Cheng, M.-M., Zhao, K., Zhang, X.-Y., Yang, M.-H., and Torr, P. (2021). Res2Net: A New Multi-Scale Backbone Architecture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(2):652–662.

[53] Garcia-Romero, D. and Espy-Wilson, C. Y. (2011). Analysis of i-vector length normalization in speaker recognition systems. In *Interspeech 2011*, pages 249–252.

[54] Garcia-Romero, D., Sell, G., and Mccree, A. (2020). MagNetO: X-vector Magnitude Estimation Network plus Offset for Improved Speaker Recognition. In *The Speaker and Language Recognition Workshop (Odyssey 2020)*, pages 1–8.

[55] Garcia-Romero, D., Snyder, D., Sell, G., Povey, D., and McCree, A. (2017). Speaker diarization using deep neural network embeddings. In *ICASSP 2017 - 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4930–4934.

[56] Gauvain, J.-L. and Lee, C.-H. (1994). Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Transactions on Speech and Audio Processing*, 2(2):291–298.

[57] Gelly, G. and Gauvain, J.-L. (2018). Optimization of RNN-Based Speech Activity Detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(3):646–656.

[58] Ghalehjegh, S. H. and Rose, R. C. (2015). Deep bottleneck features for i-vector based text-independent speaker verification. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 555–560.

[59] Graff, D., Canavan, A., and Zipperlen, G. (1998). Switchboard-2 Phase I, LDC98S75.

[60] Graff, D., Miller, D., and Walker, K. (2002). Switchboard-2 Phase III Audio, LDC2002S06.

[61] Graff, D., Walker, K., and Canavan, A. (1999). Switchboard-2 Phase II, LDC99S79.

[62] Graff, D., Walker, K., and Miller, D. (2001). Switchboard Cellular Part 1 Audio, LDC2001S13.

[63] Graff, D., Walker, K., and Miller, D. (2004). Switchboard Cellular Part 2 Audio, LDC2004S07.

[64] Greenberg, C. S., Mason, L. P., Sadjadi, S. O., and Reynolds, D. A. (2020). Two decades of speaker recognition evaluation at the national institute of standards and technology. *Computer Speech Language*, 60:101032.

[65] Grezl, F., Karafiat, M., Kontar, S., and Cernocky, J. (2007). Probabilistic and Bottle-Neck Features for LVCSR of Meetings. In *ICASSP 2007 - 2007 IEEE International Conference on Acoustics, Speech and Signal Processing - (ICASSP)*, volume 4, pages IV–757–IV–760.

[66] Gulati, A., Qin, J., Chiu, C.-C., Parmar, N., Zhang, Y., Yu, J., Han, W., Wang, S., Zhang, Z., Wu, Y., and Pang, R. (2020). Conformer: Convolution-augmented Transformer for Speech Recognition. In *Interspeech 2020*, pages 5036–5040.

[67] Hansen, J. H. and Hasan, T. (2015). Speaker Recognition by Machines and Humans: A tutorial review. *IEEE Signal Processing Magazine*, 32(6):74–99.

[68] Hatch, A., Kajarekar, S., and Stolcke, A. (2006). Within-class covariance normalization for SVM-based speaker recognition. In *INTERSPEECH 2006 and 9th International Conference on Spoken Language Processing, INTERSPEECH 2006 - ICSLP*, volume 3.

[69] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

[70] Heigold, G., Moreno, I., Bengio, S., and Shazeer, N. (2016). End-to-end text-dependent speaker verification. In *ICASSP 2016 - 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5115–5119.

[71] Hermansky, H. (1990). Perceptual linear predictive (PLP) analysis of speech. *the Journal of the Acoustical Society of America*, 87(4):1738–1752.

[72] Horiguchi, S., Fujita, Y., Watanabe, S., Xue, Y., and Garcia, P. (2022a). Encoder-Decoder Based Attractors for End-to-End Neural Diarization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:1493–1507.

[73] Horiguchi, S., Fujita, Y., Watanabe, S., Xue, Y., and Nagamatsu, K. (2020). End-to-End Speaker Diarization for an Unknown Number of Speakers with Encoder-Decoder Based Attractors. In *Proc. Interspeech 2020*, pages 269–273.

[74] Horiguchi, S., Watanabe, S., García, P., Takashima, Y., and Kawaguchi, Y. (2022b). Online Neural Diarization of Unlimited Numbers of Speakers Using Global and Local Attractors. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 31:706–720.

[75] Horiguchi, S., Watanabe, S., García, P., Xue, Y., Takashima, Y., and Kawaguchi, Y. (2021). Towards Neural Diarization for Unlimited Numbers of Speakers Using Global and Local Attractors. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 98–105.

[76] Hu, J., Shen, L., and Sun, G. (2018). Squeeze-and-Excitation Networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7132–7141.

[77] Huang, J. and Bocklet, T. (2019). Intel Far-Field Speaker Recognition System for VOiCES Challenge 2019. In *Proc. Interspeech 2019*, pages 2473–2477.

[78] Huang, Z., Wang, S., and Yu, K. (2018). Angular Softmax for Short-Duration Text-independent Speaker Verification. In *Interspeech 2018*, pages 3623–3627.

[79] Hughes, T. and Mierle, K. (2013). Recurrent neural networks for voice activity detection. In *ICASSP 2013 - 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7378–7382.

[80] Huijbregts, M., van Leeuwen, D. A., and de Jong, F. M. G. (2009). The majority wins: a method for combining speaker diarization systems. In *Interspeech 2009*, pages 924–927.

[81] Ioffe, S. (2006). Probabilistic Linear Discriminant Analysis. In Leonardis, A., Bischof, H., and Pinz, A., editors, *Computer Vision – ECCV 2006*, pages 531–542, Berlin, Heidelberg. Springer Berlin Heidelberg.

[82] Istrate, D., Fredouille, C., Meignier, S., Besacier, L., and Bonastre, J. F. (2006). NIST RT'05S Evaluation: Pre-processing Techniques and Speaker Diarization on Multiple Microphone Meetings. In Renals, S. and Bengio, S., editors, *Machine Learning for Multimodal Interaction*, pages 428–439, Berlin, Heidelberg. Springer Berlin Heidelberg.

[83] Jeancolas, L., Petrovska-Delacrétaz, D., Mangone, G., Benkelfat, B.-E., Corvol, J.-C., Vidailhet, M., Lehéricy, S., and Benali, H. (2021). X-vectors: new quantitative biomarkers for early Parkinson's disease detection from speech. *Frontiers in Neuroinformatics*, 15:578369.

[84] Jin, Z., Yang, Y., Shi, M., Kang, W., Yang, X., Yao, Z., Kuang, F., Guo, L., Meng, L., Lin, L., et al. (2024). LibriheavyMix: A 20,000-Hour Dataset for Single-Channel Reverberant Multi-Talker Speech Separation, ASR and Speaker Diarization. In *Proc. Interspeech 2024*, pages 702–706.

[85] Kalda, J., Pagés, C., Marxer, R., Alumäe, T., and Bredin, H. (2024). PixIT: Joint Training of Speaker Diarization and Speech Separation from Real-world Multi-speaker Recordings. In *The Speaker and Language Recognition Workshop (Odyssey 2024)*, pages 115–122, Quebec City, France. ISCA.

[86] Kalluri, S. B., Singh, P., Roy Chowdhuri, P., Kulkarni, A., Baghel, S., Hegde, P., Sontakke, S., K T, D., Prasanna, S. M., Vijayasenan, D., and Ganapathy, S. (2024). The Second DISPLACE Challenge: DIarization of SPeaker and LAnguage in Conversational Environments. In *Interspeech 2024*, pages 1630–1634.

[87] Kataria, S., Nidadavolu, P. S., Villalba, J., Chen, N., Garcia-Perera, P., and Dehak, N. (2020). Feature Enhancement with Deep Feature Losses for Speaker Verification. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7584–7588.

[88] Kenny, P. (2010). Bayesian Speaker Verification with Heavy-Tailed Priors. In *The Speaker and Language Recognition Workshop (Odyssey 2010)*, page paper 14.

[89] Kenny, P. and Dumouchel, P. (2004). Experiments in speaker verification using factor analysis likelihood ratios. In *The Speaker and Language Recognition Workshop (Odyssey 2004)*, pages 219–226.

[90] Kenny, P., Mihoubi, M., and Dumouchel, P. (2003). New MAP estimators for speaker recognition. In *8th European Conference on Speech Communication and Technology (Eurospeech 2003)*, pages 2961–2964.

[91] Kenny, P., Ouellet, P., Dehak, N., Gupta, V., and Dumouchel, P. (2008). A Study of Interspeaker Variability in Speaker Verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(5):980–988.

[92] Khokhlov, Y., Zatvornitskiy, A., Medennikov, I., Sorokin, I., Prisyach, T., Romanenko, A., Mitrofanov, A., Bataev, V., Andrusenko, A., Korenevskaya, M., and Petrov, O. (2019). R-Vectors: New Technique for Adaptation to Room Acoustics. In *Interspeech 2019*, pages 1243–1247.

[93] Kinoshita, K., Delcroix, M., and Iwata, T. (2022a). Tight Integration Of Neural- And Clustering-Based Diarization Through Deep Unfolding Of Infinite Gaussian Mixture Model. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8382–8386.

[94] Kinoshita, K., Delcroix, M., and Tawara, N. (2021a). Advances in Integration of End-to-End Neural and Clustering-Based Diarization for Real Conversational Speech. In *Proc. Interspeech 2021*, pages 3565–3569.

[95] Kinoshita, K., Delcroix, M., and Tawara, N. (2021b). Integrating end-to-end neural and clustering-based diarization: Getting the best of both worlds. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7198–7202. IEEE.

[96] Kinoshita, K., von Neumann, T., Delcroix, M., Boeddeker, C., and Haeb-Umbach, R. (2022b). Utterance-by-utterance overlap-aware neural diarization with Graph-PIT. In *Proc. Interspeech 2022*, pages 1486–1490.

[97] Kuhn, H. W. (1955). The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97.

[98] Landini, F., Profant, J., Diez, M., and Burget, L. (2022). Bayesian HMM clustering of x-vector sequences (VBx) in speaker diarization: Theory, implementation and analysis on standard tasks. *Computer Speech & Language*, 71:101254.

[99] Landini, F. N. (2024). *FROM MODULAR TO END-TO-END SPEAKER DIARIZATION*. PhD thesis, Brno University Of Technology.

[100] Landini, F. N., Lozano Díez, A., Diez Sánchez, M., and Burget, L. (2022). From Simulated Mixtures to Simulated Conversations as Training Data for End-to-End Neural Diarization. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, volume 2022, pages 5095–5099, Incheon. International Speech Communication Association.

[101] Lei, Y., Ferrer, L., McLaren, M., and Scheffer, N. (2014a). A deep neural network speaker verification system targeting microphone speech. In *Interspeech 2014*, pages 681–685.

[102] Lei, Y., Scheffer, N., Ferrer, L., and McLaren, M. (2014b). A novel scheme for speaker recognition using a phonetically-aware deep neural network. In *ICASSP 2014 - 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1695–1699.

[103] Leung, T.-Y. and Samarakoon, L. (2021). Robust End-to-End Speaker Diarization with Conformer and Additive Margin Penalty. In *Proc. Interspeech 2021*, pages 3575–3579.

[104] Lin, Q., Cai, W., Yang, L., Wang, J., Zhang, J., and Li, M. (2020). DIHARD II is Still Hard: Experimental Results and Discussions from the DKU-LENOVO Team. In *The Speaker and Language Recognition Workshop (Odyssey 2020)*, pages 102–109.

[105] Lin, Q., Yin, R., Li, M., Bredin, H., and Barras, C. (2019a). LSTM Based Similarity Measurement with Spectral Clustering for Speaker Diarization. In *Interspeech 2019*, pages 366–370.

[106] Lin, R., Costello, C., Jankowski, C., and Mruthyunjaya, V. (2019b). Optimizing Voice Activity Detection for Noisy Conditions. In *Interspeech 2019*, pages 2030–2034.

[107] Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., and Song, L. (2017). SphereFace: Deep Hypersphere Embedding for Face Recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[108] Liu, Y. C., Han, E., Lee, C., and Stolcke, A. (2021). End-to-End Neural Diarization: From Transformer to Conformer. In *Proc. Interspeech 2021*, pages 3081–3085.

[109] Luo, Y., Chen, Z., and Yoshioka, T. (2020). 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.

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

[111] Maiti, S., Ueda, Y., Watanabe, S., Zhang, C., Yu, M., Zhang, S.-X., and Xu, Y. (2023). EEND-SS: Joint End-to-End Neural Speaker Diarization and Speech Separation for Flexible Number of Speakers. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 480–487.

[112] Mak, M.-W. and Yu, H.-B. (2014). A study of voice activity detection techniques for NIST speaker recognition evaluations. *Computer Speech & Language*, 28(1):295–313.

[113] Martin, A., Doddington, G., Kamm, T., Ordowski, M., and Przybocki, M. (1997). The DET curve in assessment of detection task performance. In *5th European Conference on Speech Communication and Technology (Eurospeech 1997)*, pages 1895–1898.

[114] Matějka, P., Glembek, O., Novotný, O., Plchot, O., Grézl, F., Burget, L., and Cernocký, J. H. (2016). Analysis of DNN approaches to speaker identification. In *ICASSP 2016 - 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5100–5104.

[115] Matějka, P., Novotný, O., Plchot, O., Burget, L., Sánchez, M. D., and Černocký, J. (2017). Analysis of Score Normalization in Multilingual Speaker Recognition. In *Interspeech 2017*, pages 1567–1571.

[116] Matějka, P., Plchot, O., Glembek, O., Burget, L., Rohdin, J., Zeinali, H., Mošner, L., Silnova, A., Novotný, O., Diez, M., and "Honza" Černocký, J. (2020). 13 years of speaker recognition research at BUT, with longitudinal analysis of NIST SRE. *Computer Speech & Language*, 63:101035.

[117] McLaren, M., Ferrer, L., Castan, D., and Lawson, A. (2016). The Speakers in the Wild (SITW) Speaker Recognition Database. In *Interspeech 2016*, pages 818–822.

[118] Medennikov, I., Korenevsky, M., Prisyach, T., Khokhlov, Y., Korenevskaya, M., Sorokin, I., Timofeeva, T., Mitrofanov, A., Andrusenko, A., Podluzhny, I., et al. (2020). Target-Speaker Voice Activity Detection: A Novel Approach for Multi-Speaker Diarization in a Dinner Party Scenario. *Proc. Interspeech 2020*, pages 274–278.

[119] Milner, R. and Hain, T. (2016). DNN-Based Speaker Clustering for Speaker Diarisation. In *Interspeech 2016*, pages 2185–2189.

[120] Moro-Velazquez, L., Villalba, J., and Dehak, N. (2020). Using X-Vectors to Automatically Detect Parkinson's Disease from Speech. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1155–1159.

[121] Nagrani, A., Chung, J. S., Huh, J., Brown, A., Coto, E., Xie, W., McLaren, M., Reynolds, D. A., and Zisserman, A. (2020). VoxSRC 2020: The Second VoxCeleb Speaker Recognition Challenge. *arXiv preprint arXiv:2012.06867*.

[122] Nagrani, A., Chung, J. S., Xie, W., and Zisserman, A. (2019). Voxceleb: Large-scale speaker verification in the wild. *Computer Science and Language*.

[123] Nagrani, A., Chung, J. S., and Zisserman, A. (2017). VoxCeleb: A Large-Scale Speaker Identification Dataset. *Proc. Interspeech 2017*, pages 2616–2620.

[124] Nandwana, M. K., van Hout, J., Richey, C., McLaren, M., Barrios, M. A., and Lawson, A. (2019). The VOiCES from a Distance Challenge 2019. In *Interspeech 2019*, pages 2438–2442.

[125] Ning, H., Liu, M., Tang, H., and Huang, T. S. (2006). A spectral clustering approach to speaker diarization. In *Interspeech 2006*, pages paper 1607–Thu1A1O.1.

[126] NIST, A. (2010). The NIST year 2010 speaker recognition evaluation plan. *URL http://www. nist. gov/itl/iad/mig/upload/NIST_SRE10_evalplan-r6. pdf.*

[127] NIST Multimodal Information Group (2006a). 2004 NIST Speaker Recognition Evaluation, LDC2006S44.

[128] NIST Multimodal Information Group (2006b). 2005 NIST Speaker Recognition Evaluation Training Data, LDC2011S01.

[129] NIST Multimodal Information Group (2011a). 2005 NIST Speaker Recognition Evaluation Test Data, LDC2011S04.

[130] NIST Multimodal Information Group (2011b). 2006 NIST Speaker Recognition Evaluation Test Set Part 1, LDC2011S10.

[131] NIST Multimodal Information Group (2011c). 2006 NIST Speaker Recognition Evaluation Training Set, LDC2011S09.

[132] NIST Multimodal Information Group (2011d). 2008 NIST Speaker Recognition Evaluation Test Set, LDC2011S08.

[133] NIST Multimodal Information Group (2011e). 2008 NIST Speaker Recognition Evaluation Training Set Part 1, LDC2011S05.

[134] NIST Multimodal Information Group (2012). 2006 NIST Speaker Recognition Evaluation Test Set Part 2, LDC2012S01.

[135] Novoselov, S., Gusev, A., Ivanov, A., Pekhovsky, T., Shulipa, A., Avdeeva, A., Gorlanov, A., and Kozlov, A. (2019). Speaker Diarization with Deep Speaker Embeddings for DIHARD Challenge II. In *Interspeech 2019*, pages 1003–1007.

[136] Novoselov, S., Shulipa, A., Kremnev, I., Kozlov, A., and Shchemelinin, V. (2018). On deep speaker embeddings for text-independent speaker recognition . In *The Speaker and Language Recognition Workshop (Odyssey 2018)*, pages 378–385.

[137] Okabe, K., Koshinaka, T., and Shinoda, K. (2018). Attentive Statistics Pooling for Deep Speaker Embedding. In *Interspeech 2018*, pages 2252–2256.

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

[139] Pappagari, R., Cho, J., Moro-Velázquez, L., and Dehak, N. (2020). Using State of the Art Speaker Recognition and Natural Language Processing Technologies to Detect Alzheimer's Disease and Assess its Severity. In *Interspeech 2020*, pages 2177–2181.

[140] Park, T. J., Kanda, N., Dimitriadis, D., Han, K. J., Watanabe, S., and Narayanan, S. (2022). A review of speaker diarization: Recent advances with deep learning. *Computer Speech & Language*, 72:101317.

[141] Peddinti, V., Povey, D., and Khudanpur, S. (2015). A time delay neural network architecture for efficient modeling of long temporal contexts. In *Interspeech 2015*, pages 3214–3218.

[142] Plchot, O., Matsoukas, S., Matějka, P., Dehak, N., Ma, J., Cumani, S., Glembek, O., Hermansky, H., Mallidi, S., Mesgarani, N., Schwartz, R., Soufifar, M., Tan, Z., Thomas, S., Zhang, B., and Zhou, X. (2013). Developing a speaker identification system for the DARPA RATS project. In *ICASSP 2013 - 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6768–6772.

[143] Pompili, A., Rolland, T., and Abad, A. (2020). The INESC-ID Multi-Modal System for the ADReSS 2020 Challenge. In *Interspeech 2020*, pages 2202–2206.

[144] Povey, D., Cheng, G., Wang, Y., Li, K., Xu, H., Yarmohammadi, M., and Khudanpur, S. (2018). Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks. In *Proc. Interspeech 2018*, pages 3743–3747.

[145] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., et al. (2011). The Kaldi speech recognition toolkit. Technical report, IEEE Signal Processing Society.

[146] Prince, S. and Elder, J. (2007). Probabilistic Linear Discriminant Analysis for Inferences About Identity. In *IEEE 11th International Conference on Computer Vision*, pages 1–8.

[147] Przybocki, M. and Alvin, M. (2001). 2000 NIST Speaker Recognition Evaluation LDC2001S97. Web Download. Philadelphia: Linguistic Data Consortium.

[148] Qin, X., Li, M., Bu, H., Rao, W., Das, R. K., Narayanan, S., and Li, H. (2020). The INTERSPEECH 2020 Far-Field Speaker Verification Challenge. In *Interspeech 2020*, pages 3456–3460.

[149] Raj, D., Denisov, P., Chen, Z., Erdogan, H., Huang, Z., He, M., Watanabe, S., Du, J., Yoshioka, T., Luo, Y., et al. (2021a). Integration of speech separation, diarization, and recognition for multi-speaker meetings: System description, comparison, and analysis. In *2021 IEEE spoken language technology workshop (SLT)*, pages 897–904. IEEE.

[150] Raj, D., Huang, Z., and Khudanpur, S. (2021b). Multi-Class Spectral Clustering with Overlaps for Speaker Diarization. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 582–589.

[151] Raj, D., Paola Garcia-Perera, L., Huang, Z., Watanabe, S., Povey, D., Stolcke, A., and Khudanpur, S. (2021c). DOVER-Lap: A Method for Combining Overlap-Aware Diarization Outputs. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 881–888.

[152] Reynolds, D. (1994). Experimental evaluation of features for robust speaker identification. *IEEE Transactions on Speech and Audio Processing*, 2(4):639–643.

[153] Reynolds, D. and Rose, R. (1995). Robust text-independent speaker identification using Gaussian mixture speaker models. *IEEE Transactions on Speech and Audio Processing*, 3(1):72–83.

[154] Reynolds, D. A. (1997). Comparison of background normalization methods for text-independent speaker verification. In *5th European Conference on Speech Communication and Technology (Eurospeech 1997)*, pages 963–966.

[155] Reynolds, D. A., Quatieri, T. F., and Dunn, R. B. (2000). Speaker verification using adapted Gaussian mixture models. *Digital signal processing*, 10(1-3):19–41.

[156] Rix, A., Beerends, J., Hollier, M., and Hekstra, A. (2001). Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings*, volume 2, pages 749–752 vol.2.

[157] Roux, J. L., Wisdom, S., Erdogan, H., and Hershey, J. R. (2019). SDR – Half-baked or Well Done? In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 626–630.

[158] Ryant, N., Church, K., Cieri, C., Cristia, A., Du, J., Ganapathy, S., and Liberman, M. (2018). First DIHARD Challenge Evaluation Plan. Technical report, Linguistic Data Consortium.

[159] Ryant, N., Church, K., Cieri, C., Cristia, A., Du, J., Ganapathy, S., and Liberman, M. (2019a). Second DIHARD Challenge Evaluation Plan v1.1. https://dihardchallenge.github.io/dihard2/docs/second_dihard_eval_plan_v1.1.pdf.

[160] Ryant, N., Church, K., Cieri, C., Cristia, A., Du, J., Ganapathy, S., and Liberman, M. (2019b). The Second DIHARD Diarization Challenge: Dataset, Task, and Baselines. In *Interspeech 2019*, pages 978–982.

[161] Ryant, N., Singh, P., Krishnamohan, V., Varma, R., Church, K., Cieri, C., Du, J., Ganapathy, S., and Liberman, M. (2020). The third DIHARD diarization challenge. *arXiv preprint arXiv:2012.01477*.

[162] Rybicka, M., Kacprzak, S., Witkowski, M., and Kowalczyk, K. (2020). Description of the DSP AGH systems for the SdSV Challenge. Technical report.

[163] Rybicka, M., Villalba, J., Żelasko, P., Dehak, N., and Kowalczyk, K. (2021). Spine2Net: SpineNet with Res2Net and Time-Squeeze-and-Excitation Blocks for Speaker Recognition. In *Interspeech 2021*, pages 496–500.

[164] Sarma, M., Ghahremani, P., Povey, D., Goel, N. K., Sarma, K. K., and Dehak, N. (2018). Emotion Identification from Raw Speech Signals Using DNNs. In *Interspeech 2018*, pages 3097–3101.

[165] Sell, G. and Garcia-Romero, D. (2014). Speaker diarization with plda i-vector scoring and unsupervised calibration. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 413–417.

[166] Sell, G. and Garcia-Romero, D. (2015). Diarization resegmentation in the factor analysis subspace. In *ICASSP 2015 - 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4794–4798.

[167] Sell, G., Snyder, D., McCree, A., Garcia-Romero, D., Villalba, J., Maciejewski, M., Manohar, V., Dehak, N., Povey, D., Watanabe, S., and Khudanpur, S. (2018). Diarization is Hard: Some Experiences and Lessons Learned for the JHU Team in the Inaugural DIHARD Challenge. In *Interspeech 2018*, pages 2808–2812.

[168] Shum, S., Dehak, N., and Glass, J. (2012). On the use of spectral and iterative methods for speaker diarization. In *Interspeech 2012*, pages 482–485.

[169] Shum, S. H., Dehak, N., Dehak, R., and Glass, J. R. (2013). Unsupervised Methods for Speaker Diarization: An Integrated and Iterative Approach. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(10):2015–2028.

[170] Siegler, M. A., Jain, U., Raj, B., and Stern, R. M. (1997). Automatic Segmentation, Classification and Clustering of Broadcast News Audio. In *Proceedings of the DARPA Speech Recognition Workshop*, Chantilly, Virginia.

[171] Snyder, D. (2020). *X-VECTORS: ROBUST NEURAL EMBEDDINGS FOR SPEAKER RECOGNITION*. PhD thesis, The Johns Hopkins University.

[172] Snyder, D., Garcia-Romero, D., McCree, A., Sell, G., Povey, D., and Khudanpur, S. (2018a). Spoken Language Recognition using X-vectors. In *The Speaker and Language Recognition Workshop (Odyssey 2018)*, pages 105–111.

[173] Snyder, D., Garcia-Romero, D., Povey, D., and Khudanpur, S. (2017). Deep Neural Network Embeddings for Text-Independent Speaker Verification. In *Interspeech 2017*, pages 999–1003.

[174] Snyder, D., Garcia-Romero, D., Sell, G., McCree, A., Povey, D., and Khudanpur, S. (2019a). Speaker Recognition for Multi-speaker Conversations Using X-vectors. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5796–5800.

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

[176] Snyder, D., Ghahremani, P., Povey, D., Garcia-Romero, D., Carmiel, Y., and Khudanpur, S. (2016). Deep neural network-based speaker embeddings for end-to-end speaker verification. In *2016 IEEE Spoken Language Technology Workshop (SLT)*, pages 165–170.

[177] Snyder, D., Villalba, J., Chen, N., Povey, D., Sell, G., Dehak, N., and Khudanpur, S. (2019b). The JHU Speaker Recognition System for the VOiCES 2019 Challenge. In *Proc. Interspeech 2019*, pages 2468–2472.

[178] Srikanth Raj Chetupalli and Emanuël Habets (2022). Speech Separation for an Unknown Number of Speakers Using Transformers With Encoder-Decoder Attractors. In *Interspeech 2022*, pages 5393–5397.

[179] Stolcke, A. and Yoshioka, T. (2019). Dover: A Method for Combining Diarization Outputs. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 757–763.

[180] Subakan, C., Ravanelli, M., Cornell, S., Bronzi, M., and Zhong, J. (2021). Attention is all you need in speech separation. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 21–25. IEEE.

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

[182] Taherian, H. and Wang, D. (2024). Multi-Channel Conversational Speaker Separation via Neural Diarization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:2467–2476.

[183] Tan, L. N., Borgstrom, B. J., and Alwan, A. (2010). Voice activity detection using harmonic frequency components in likelihood ratio test. In *ICASSP 2010 - 2010 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4466–4469.

[184] Variani, E., Lei, X., McDermott, E., Moreno, I. L., and Gonzalez-Dominguez, J. (2014). Deep neural networks for small footprint text-dependent speaker verification. In *ICASSP 2014 - 2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4052–4056. IEEE.

[185] Villalba, J., Borgstrom, B. J., Kataria, S., Rybicka, M., Castillo, C. D., Cho, J., García-Perera, L. P., Torres-Carrasquillo, P. A., and Dehak, N. (2022). Advances in Cross-Lingual and Cross-Source Audio-Visual Speaker Recognition: The JHU-MIT System for NIST SRE21. In *The Speaker and Language Recognition Workshop (Odyssey 2022)*, pages 213–220.

[186] Villalba, J., Chen, N., Snyder, D., Garcia-Romero, D., McCree, A., Sell, G., Borgstrom, J., García-Perera, L. P., Richardson, F., Dehak, R., Torres-Carrasquillo, P. A., and Dehak, N. (2020a). State-of-the-art speaker recognition with neural network embeddings in NIST SRE18 and Speakers in the Wild evaluations. *Computer Speech & Language*, 60:101026.

[187] Villalba, J., Garcia-Romero, D., Chen, N., Sell, G., Borgstrom, B., McCree, A., Garcia, P., Kataria, S., Nidadavolu, P., Torres-Carrasquiilo, P., and Dehak, N. (2020b). Advances in Speaker Recognition for Telephone and Audio-Visual Data: the JHU-MIT Submission for NIST SRE19. In *Proc. Odyssey 2020*, pages 273–280.

[188] Vincent, E., Gribonval, R., and Fevotte, C. (2006). Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(4):1462–1469.

[189] von Neumann, T., Kinoshita, K., Boeddeker, C., Delcroix, M., and Haeb-Umbach, R. (2021). Graph-PIT: Generalized Permutation Invariant Training for Continuous Separation of Arbitrary Numbers of Speakers. In *Interspeech 2021*, pages 3490–3494.

[190] Wang, F., Cheng, J., Liu, W., and Liu, H. (2018a). Additive Margin Softmax for Face Verification. *IEEE Signal Processing Letters*, 25(7):926–930.

[191] Wang, Q., Downey, C., Wan, L., Mansfield, P. A., and Moreno, I. L. (2018b). Speaker Diarization with LSTM. In *ICASSP 2018 - 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5239–5243.

[192] Wang, Z., Yao, K., Li, X., and Fang, S. (2020). Multi-Resolution Multi-Head Attention in Deep Speaker Embedding. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6464–6468.

[193] Watanabe, S., Mandel, M., Barker, J., Vincent, E., Arora, A., Chang, X., Khudanpur, S., Manohar, V., Povey, D., Raj, D., Snyder, D., Subramanian, A. S., Trmal, J., Yair, B. B., Boeddeker, C., Ni, Z., Fujita, Y., Horiguchi, S., Kanda, N., Yoshioka, T., and Ryant, N. (2020). CHiME-6 Challenge: Tackling Multispeaker Speech Recognition for Unsegmented Recordings. In *6th International Workshop on Speech Processing in Everyday Environments (CHiME 2020)*, pages 1–7.

[194] Wichern, G., Antognini, J., Flynn, M., Zhu, L. R., McQuinn, E., Crow, D., Manilow, E., and Roux, J. L. (2019). WHAM!: Extending Speech Separation to Noisy Environments. In *Interspeech 2019*, pages 1368–1372.

[195] Witkowski, M., Rybicka, M., and Kowalczyk, K. (2019). Speaker Recognition from Distance Using X-Vectors with Reverberation-Robust Features. In *2019 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*, pages 235–240.

[196] Witkowski, M., Rybicka, M., and Kowalczyk, K. (2021). Sparse Linear Prediction-based Dereverberation for Signal Enhancement in Distant Speaker Verification. In *2021 29th European Signal Processing Conference (EUSIPCO)*, pages 461–465.

[197] Xiang, X., Wang, S., Huang, H., Qian, Y., and Yu, K. (2019). Margin Matters: Towards More Discriminative Deep Neural Network Embeddings for Speaker Recognition. In *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1652–1656.

[198] Xiao, X., Kanda, N., Chen, Z., Zhou, T., Yoshioka, T., Chen, S., Zhao, Y., Liu, G., Wu, Y., Wu, J., Liu, S., Li, J., and Gong, Y. (2020). Microsoft Speaker Diarization System for the VoxCeleb Speaker Recognition Challenge 2020.

[199] Xie, W., Nagrani, A., Chung, J. S., and Zisserman, A. (2019). Utterance-level Aggregation for Speaker Recognition in the Wild. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5791–5795.

[200] Yu, C. and Hansen, J. H. L. (2017). Active Learning Based Constrained Clustering For Speaker Diarization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(11):2188–2198.

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

[202] Yu, Y., Park, D., and Kook Kim, H. (2022). Auxiliary Loss of Transformer with Residual Connection for End-to-End Speaker Diarization. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8377–8381.

[203] Yu, Y.-Q., Fan, L., and Li, W.-J. (2019). Ensemble Additive Margin Softmax for Speaker Verification. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6046–6050.

[204] Zeinali, H., Lee, K. A., Alam, J., and Burget, L. (2020). SdSV Challenge 2020: Large-Scale Evaluation of Short-Duration Speaker Verification. In *Interspeech 2020*, pages 731–735.

[205] Zeinali, H., Wang, S., Silnova, A., Matějka, P., and Plchot, O. (2019). But system description to voxceleb speaker recognition challenge 2019. *arXiv preprint arXiv:1910.12592*.

[206] Zhang, A., Wang, Q., Zhu, Z., Paisley, J., and Wang, C. (2019a). Fully Supervised Speaker Diarization. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6301–6305.

[207] Zhang, C., Koishida, K., and Hansen, J. H. L. (2018). Text-Independent Speaker Verification Based on Triplet Convolutional Neural Network Embeddings. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(9):1633–1644.

[208] Zhang, X., Zhao, R., Qiao, Y., Wang, X., and Li, H. (2019b). AdaCos: Adaptively Scaling Cosine Logits for Effectively Learning Deep Face Representations. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[209] Zhu, Y., Ko, T., Snyder, D., Mak, B., and Povey, D. (2018). Self-Attentive Speaker Embeddings for Text-Independent Speaker Verification. In *Interspeech 2018*, pages 3573–3577.

[210] Zigel, Y. and Wasserblat, M. (2006). How to Deal with Multiple-Targets in Speaker Identification Systems? In *2006 IEEE Odyssey - The Speaker and Language Recognition Workshop*, pages 1–7.

Publication Series - Full Texts

This Appendix contains the full texts of the publications that form this dissertation. The publications are arranged chronologically.

I On Parameter Adaptation in Softmax-Based Cross-Entropy Loss for Improved Convergence Speed and Accuracy in DNN-Based Speaker Recognition

M. Rybicka and K. Kowalczyk "*On Parameter Adaptation in Softmax-Based Cross-Entropy Loss for Improved Convergence Speed and Accuracy in DNN-Based Speaker Recognition*", Interspeech, Shanghai, China, 2020.

INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

On Parameter Adaptation in Softmax-based Cross-Entropy Loss for Improved Convergence Speed and Accuracy in DNN-based Speaker Recognition

Magdalena Rybicka and Konrad Kowalczyk

AGH University of Science and Technology
Department of Electronics
30-059 Krakow, Poland

mrybicka@agh.edu.pl, konrad.kowalczyk@agh.edu.pl

Abstract

In various classification tasks the major challenge is in generating discriminative representation of classes. By proper selection of deep neural network (DNN) loss function we can encourage it to produce embeddings with increased inter-class separation and smaller intra-class distances. In this paper, we develop softmax-based cross-entropy loss function which adapts its parameters to the current training phase. The proposed solution improves accuracy up to 24% in terms of Equal Error Rate (EER) and minimum Detection Cost Function (minDCF). In addition, our proposal also accelerates network convergence compared with other state-of-the-art softmax-based losses. As an additional contribution of this paper, we adopt and subsequently modify the ResNet DNN structure for the speaker recognition task. The proposed ResNet network achieves relative gains of up to 32% and 15% in terms of EER and minDCF respectively, compared with the well-established Time Delay Neural Network (TDNN) architecture for x-vector extraction.

Index Terms: speaker recognition, deep neural networks, softmax activation functions, speaker embedding, ResNet

1. Introduction

Recently we have seen a rapid increase in popularity of speaker modeling using deep neural networks (DNNs) [1, 2, 3]. State-of-the-art solution consist in extracting a speaker embedding from the Time Delay Neural Network (TDNN), which is commonly referred to as the x-vector [2]. Extensions of the basic TDNN structure have been presented e.g. in [4, 5], while different network structures for the extraction of speaker embeddings have very recently been proposed e.g. in [1, 3, 6, 7, 8, 9]. Finding an appropriate network structure which facilitates notable improvement in speaker recognition performance is thus still an on-going research topic.

In speaker recognition, it is common to use a cross-entropy loss function with softmax activations in the last DNN layer [1, 4]. This loss function is also widely used in other tasks such as image recognition [10] or speech emotion recognition [11]. Recent modifications to such loss functions include increasing inter-class separation by an introduction of various types of margins in the angular functions [12, 13, 14]. They have been successfully applied in speaker recognition e.g. in [15, 16]. The convergence of the training process and the resulting model performance strongly depend on the selection of hyperparameters of the modified loss function, which often need to be tuned by repeating the training with different hyperparameter values. To address this issue, in [17] a cosine-based activation function called AdaCos has been proposed for the face recognition application, which adapts the scale parameter in angular softmax representation to improve the training effectiveness.

In this paper, we aim to develop a softmax-based cross-entropy loss function which adapts its hyperparameters so that they strengthen supervision at different neural network training phases. The proposed approach allows to adapt the scale and additive angular margin parameters in joint softmax-based cross-entropy loss function, resulting in a notable improvement in convergence speed of the network training and in speaker recognition accuracy. In addition, we propose a modification of the Residual Network (ResNet) [18] architecture which shows significant improvements in accuracy over the standard TDNN architecture. In evaluations, we compare the proposed parameter adaptation (ParAda) in softmax-based loss function in terms of convergence speed and accuracy for two speaker recognition systems based on TDNN and the proposed modified ResNet.

2. Softmax-based cross-entropy loss functions

In this section, we present an overview of several recently proposed modifications of a standard cross-entropy loss with softmax activation functions for DNN training, and next we propose a softmax-based loss function with adaptive parameters (ParAda) which improves discriminative capabilities of the network and accelerates its convergence. Let us first observe that the dot product between the softmax layer input vector and the weight vector can be written as $\mathbf{w}_k^T \mathbf{x}_i = \|\mathbf{w}_k\| \, \|\mathbf{x}_i\| \, \cos(\theta_{i,k})$, where $\|\mathbf{w}_k\|$ denotes the norm of the weight vector for the kth class, $k = 1, 2, ..., K$ with K being the number of speakers in the entire training set, $\|\mathbf{x}_i\|$ denotes the norm of the input vector for the ith minibatch example, $i = 1, 2, ..., N$ with N denoting the minibatch size, and $\cos(\theta_{i,k})$ denotes the cosine angular distance between vectors $\mathbf{w}_k$ and $\mathbf{x}_i$. Next, we can express the general equation for the softmax-based cross-entropy loss function as

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} \log P_{y_i} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{f_{y_i}}}{e^{f_{y_i}} + \sum\limits_{k=1, k\neq y_i}^{K} e^{f_{y_i,k}}} \,, \quad (1)$$

where y_i is the ground truth label of a training example, P_{y_i} denotes the predicted classification probability of all samples in the minibatch, while f_{y_i} and $f_{y_i,k}$ denote the target and non-target logits given respectively by

$$f_{y_i} = s(\theta_{y_i})\, \psi(\theta_{y_i}) \,, \quad (2)$$

$$f_{y_i,k} = s(\theta_{y_i})\, \cos(\theta_{y_i,k}) \,. \quad (3)$$

This research received financial support from the Foundation for Polish Science under grant number First TEAM/2017-3/23 (POIR.04.04.00-00-3FC4/17-00) which is co-financed by the European Union and was supported in part by PLGrid Infrastructure.

http://dx.doi.org/10.21437/Interspeech.2020-2264

In standard softmax-based cross-entropy loss, $\psi(\theta_{y_i})$ is defined as the cosine of the angle between the ith minibatch input vector and the weight vector corresponding to its ground truth label. Note that for the convenience of comparing various softmax modifications, in (2) and (3) we normalized the weight vectors for all classes such that $\|\mathbf{w}_k\| = 1$ and replaced the norm of an input vector for the true class $\|\mathbf{x}_i\|$ with a new scale variable $s(\theta_{y_i})$.

2.1. State-of-the-art fixed angular and scale functions

There are three types of modifications of the standard angular function $\psi(\theta_{y_i})$ in (2), namely the so-called Angular Softmax (AS) [12], Additive Angular Softmax (AAS) [13], and Additive Margin Softmax (AMS) [14], which can all be presented in a general form

$$\psi(\theta_{y_i}) = \cos(m_{\mathrm{AS}}\,\theta_{y_i} + m_{\mathrm{AAS}}) - m_{\mathrm{AMS}}, \qquad (4)$$

where m_{AS}, m_{AAS} and m_{AMS} are the real numbers for each modification. Although proper setting of these parameters has been shown to improve the accuracy of DNN-based speaker recognition [15, 16], the disadvantage of these approaches is that parameter tuning requires time-consuming repetitions of the network training. Another approach is taken in [17] in which a fixed scale function $s(\theta_{y_i}) = s_{\mathrm{Fix}}$ is given by a constant

$$s_{\mathrm{Fix}} = \sqrt{2}\,\log\,(K-1), \qquad (5)$$

which depends on the number of speakers K in the training, which allows to avoid scale parameter tuning.

2.2. Adaptation of the scaling parameter

In this section, we discuss a method to adapt the scale parameter $s(\theta_{y_i})$ depending on network convergence at current training iteration. This method is based on the recently proposed Adaptively Scaling Cosine Logits (AdaCos) introduced in [17] in the context of face recognition, which relies on adapting the scale function $s(\theta_{y_i})$ during the network training. As derived in [17], the scale adaptation (SAda) is given by

$$s_{\mathrm{Ada}}(\theta_{y_i}) = \begin{cases} \sqrt{2}\log(K-1) & \mathrm{iter} = 0 \\ \frac{\log(B_{\mathrm{SAda}})}{\cos(\min(\frac{\pi}{4},\Theta))} & \mathrm{iter} \geq 1 \end{cases} \qquad (6)$$

where $\Theta = \mathrm{median}(\theta_{y_1}, \theta_{y_2}, ..., \theta_{y_N})$ denotes the median of angles θ_{y_i} over the entire minibatch of length N, and iter denotes the iteration index. B_{SAda} denotes the summation of exponential functions of logits for all non-corresponding classes, averaged over the entire minibatch of size N, which is given by

$$B_{\mathrm{SAda}} = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1,k\neq y_i}^{K} e^{\widetilde{s}_{\mathrm{Ada}}\cos(\theta_{i,k})}, \qquad (7)$$

where $\widetilde{s}_{\mathrm{Ada}} = s_{\mathrm{Ada}}(\mathrm{iter}-1)$ denotes the scale parameter value calculated according to (6) in the previous iteration.

2.3. Adaptation of the margin-based angular function

In this section, we propose a method to adapt the margin parameter (MAda) in an angular function depending on network convergence state in the current iteration of the network training. The considered angular function is given by $\psi(\theta_{y_i}) = \cos(\theta_{y_i} + m_{\mathrm{Ada}})$, where m_{Ada} denotes the adaptive margin parameter. Since we aim to find a margin parameter which significantly changes the predicted classification probability P_{y_i} of all samples in the minibatch, similarly to [17], we calculate the point θ_0 where absolute gradient value of the predicted probability reaches the maximum value. It is found as point at which the second-order derivative of P_{y_i} is equal to zero, which yields the approximated relation for the angular function

$$\cos(\theta_0 + m_{\mathrm{Ada}}) = \frac{1}{s_m}\,\log\Big(\sum_{k=1,k\neq y_i}^{K} e^{s_m\cos(\theta_{i,k})}\Big), \qquad (8)$$

where s_m denotes the fixed scale parameter in margin adaptation and $\theta_0 \in [0, \frac{\pi}{2}]$. In order to reflect the convergence state of the network in the current minibatch, we replace θ_0 with the median of angles θ_{y_i} over the entire minibatch, i.e., $\theta_0 = \Theta = \mathrm{median}(\theta_{y_1}, \theta_{y_2}, ..., \theta_{y_N})$, which yields the following update for margin adaptation (MAda)

$$m_{\mathrm{Ada}} = \arccos\Big(\frac{1}{s_m}\log(B_{\mathrm{MAda}})\Big) - \Theta\,, \qquad (9)$$

$$B_{\mathrm{MAda}} = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1,k\neq y_i}^{K} e^{s_m\cos(\theta_{i,k})}. \qquad (10)$$

2.3.1. Annealing strategy in margin-based angular function

Similarly to the procedure presented in [15] for the stabilization of the network training for fixed margin-based softmax, the annealing strategy can also be incorporated into the proposed softmax function with adaptive margin. The angular function with margin adaptation (MAda) then takes the form

$$\psi_{\mathrm{MAda}}(\theta_{y_i}) = \frac{1}{1+\gamma}\cos(\theta_{y_i} + m_{\mathrm{Ada}}) + \frac{\gamma}{1+\gamma}\cos(\theta_{y_i}) \quad (11)$$

where $\gamma = \max\{\gamma_{min}, \gamma_b(1+\beta\cdot\mathrm{iter})^{-\alpha}\}$ where iter is training iteration, γ_{min} is the minimum value, while γ_b, β and α are hyperparamters that control the annealing speed.

2.3.2. Lower bound on the scale parameter value

By noting that $\arccos(\cdot)$ function in (9) is only defined for arguments from the range $[-1, 1]$ and that $\theta_{i,k} \in [0, \frac{\pi}{2}]$, we can find the lower bound on the value of the scale parameter in the proposed MAda approach by solving inequality

$$-1 \leq \frac{1}{s_m}\,\log\Big(\sum_{k=1,k\neq y_i}^{K} e^{s_m\cos(\theta_{i,k})}\Big) \leq 1\,. \qquad (12)$$

Assuming that all $\theta_{i,k}$ are approximately equal, we can set $\forall\, k, i\ \theta_{i,k} = \bar{\theta}$, and by replacing the values and solving (12) we obtain the lower bound on the scale parameter s_m value

$$s_m \geq [1-\cos(\bar{\theta})]^{-1}\,\log(K-1)\,. \qquad (13)$$

In an ideal case, after network training, $\bar{\theta} \to \frac{\pi}{2}$ which would yield $s_m \to \log(K-1)$; while in the worst case, $\bar{\theta} \to 0$ which would impose $s_m \to \infty$. A reasonable assumption for the practical parameter setting is to assume that $\bar{\theta} = \frac{\pi}{4}$ and take a slightly higher value of s_m than the one obtained from (13).

2.4. Proposed softmax loss with parameter adaptation

This section presents the proposed method for parameter adaptation (ParAda) in softmax-based cross-entropy loss function. We focus on adapting the scale and margin parameters which affect the shape of the predicted classification probability P_{y_i} function, namely its range and the position of an inflection point. To facilitate the training process, we take the approach of gradual increasing the network training supervision. In the initial training phase, we aim to ease learning by shifting the inflection point in $\frac{\pi}{2}$ direction so that the probabilities would be high even for the relatively high values of θ_{y_i} angles. Along with further learning, the curve adaptively shifts towards the

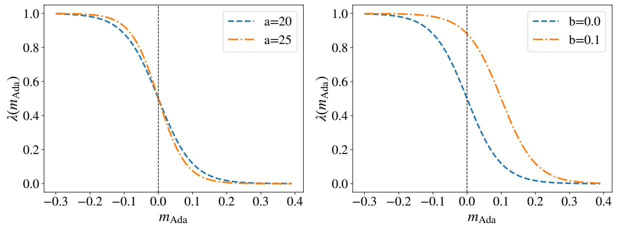


Figure 1: *The λ function with respect to the m_{Ada} margin value for $b = 0$ (left plot) and $a = 20$ (right plot).*

lower angles, thereby increasing the discriminative capability of the classification probability. We can realize it by emphasizing margin adaptation (which in an adaptive version starts from negative values), and as training progresses, more emphasis is put on the scale adaptation. To achieve this goal, we propose the following modifications of logits of the softmax-based loss function which facilitate the parameter adaptation (ParAda):

$$f_{y_i} = \lambda \cdot s_m \psi_{\mathrm{MAda}}(\theta_{y_i}) + (1-\lambda) \cdot s_{\mathrm{Ada}}(\theta_{y_i}) \cos(\theta_{y_i}), \quad (14)$$

$$f_{y_i,k} = \lambda \cdot s_m \cos(\theta_{y_i,k}) + (1-\lambda) \cdot s_{\mathrm{Ada}}(\theta_{y_i}) \cos(\theta_{y_i,k}). \quad (15)$$

A strongly desirable, integral property of ParAda is that the value of an adaptive parameter $\lambda(m_{\mathrm{Ada}})$ can be adjusted depending on the level of convergence of the trained network. The desired smooth transition between the margin adaptation (MAda) and the scale adaptation (SAda) is obtained when

$$\lambda(m_{\mathrm{Ada}}) = [1 + e^{a \cdot (m_{\mathrm{Ada}} - b)}]^{-1}, \quad (16)$$

where a and b are hyperparameters of $\lambda(m_{\mathrm{Ada}})$, whose shape for example parameter values is presented in Fig. 1. Parameter a controls the speed of switching between the two loss types, while parameter b shifts the inflection point of the function.

3. Deep neural network architectures

In this section, we outline the baseline TDNN architecture for x-vector extraction [2] and propose a set of modifications to adapt the original ResNet structure for speaker embedding extraction.

3.1. Time Delay Neural Network for x-vector extraction

The Time Delay Neural Network [2] consists of 5 delay-type layers which operate on 1D speech frames, extracting the temporal context of 15 frames. Next, the statistics pooling layer computes the mean and standard deviation of the aggregated outputs of the 5th layer for the entire utterance. The final part of the network consists of two fully connected layers followed by a softmax layer. During testing, the output of the first fully connected layer is used as the x-vector embedding.

3.2. Modified ResNet for speaker embedding extraction

In this work, we propose modifications of the Residual Network 18 (ResNet18) architecture for speaker embedding extraction with improved performance. The proposed architecture is composed of the following elements. A 2D feature vector is fed into a single 2D convolution layer with 7x7 filter size and stride of 2x2. Next, the network is composed of 4 large segments, each containing a different number of blocks which consists of 2 consecutive convolutional layers with the so-called identity shortcut connection that skips the entire block. The number of such 2-layer blocks is equal to {2, 2, 2, 2} for all segments. The first convolutional layer of a segment downsamples the input along the feature dimension by 2, while the sizes of convolutional layer outputs for each segment are respectively given as {64, 128, 256, 512}. Then the statistics pooling layer computes the mean and standard deviation in time domain of the outputs from the convolutional segments. The pooling is followed by 2 fully connected layers with output size of 512 each, which are added before the softmax classification layer. Hereafter, we will refer to this structure as modified ResNet18 (mR18).

With regard to the existing ResNet-based speaker recognition systems, the original ResNet34 and ResNet50 network structures [18] with a fully connected layer added right before the global mean pooling layer are used in [1]. An analogous solution has been presented in [19] for ResNet18 and ResNet34. In [7] all TDNN layers have been replaced with the ResNet34 residual blocks with learnable dictionary encoding (LDE) layer [20] instead of the pooling layer. In [8] a fully connected layer has been inserted after pooling operation in ResNet50 structure.

4. Experimental results and evaluation

4.1. Experiments, datasets and evaluation measures

In this section, we present the system and datasets for 2 performed experiments. The first experiment is carried out on the VoxCeleb1 corpus [21] with training part used for the DNN, LDA and PLDA training, and test part of VoxCeleb1 used for system evaluation. In Experiment 2, the entire VoxCeleb2 corpus [1] is added to the training part of VoxCeleb1 for NN training. Each training dataset is extended by data augmentations of four types: convolution with reverberation (simulated RIRs [22] from small and medium sized rooms), augmented by adding music, ambient noises, and overlapping speech of randomly selected 3-7 speakers from the MUSAN corpus [23]. The final training dataset consists of the original training data and randomly selected subset of the augmented data, which results in 348 642 utterances (1 211 speakers) for Experiment 1 and 2 276 888 utterances (7 323 speakers) for Experiment 2.

Feature extraction, LDA, and PLDA training are performed in Kaldi toolkit [24], while DNN training and embedding extraction is performed using TensorFlow implementation [15, 25]. Input features are 64-band Mel filter bank coefficients computed using frames of 25 ms length with 10 ms overlap and mean-normalized over a 3 s window. The threshold in VAD of Kaldi is set to 3.5. System backend consists of length normalization, centering with mean of training data, LDA dimensionality reduction from 512 to 200, and PLDA scoring.

In two performed experiments, we compare the performance of the existing and the proposed softmax-based loss functions using two speaker recognition systems based on the existing TDNN and the proposed mR18 architectures. In particular, in Experiment 1 we evaluate the existing standard, Additive Angular Softmax (AAS) [13] with scale equal to 30 and margin set to 0.3, Fixed Scale parameter as given by (5) [17], and adaptive scale (SAda) that is equivalent to AdaCos [17], and compare their results with the proposed adaptive margin (MAda) in the angular function as described in Sec. 2.3 and the proposed parameter adaptation (ParAda) described in Sec. 2.4 for 4 different transitions in the loss function. In Experiment 2 we evaluate only the selected 4 methods namely the standard, AAS [13], AdaCos [17], and the proposed ParAda with $a = 20$ and $b = 0.0$ on the larger training dataset. The maximum number of epochs for network training in Experiment 1 is set to 4, while in Experiment 2 it is set to 8. In order to satisfy (13), s_m is set to 30 and 35 in Experiments 1 and 2, respectively, while in both experiments the annealing function parameters are set as $\gamma_{min} = 0$, $\gamma_b = 1000$, $\beta = 0.00001$ and $\alpha = 5$.

Table 1: *Accuracy in terms of EER and minDCF, and approximate network convergence time (in Epoch) for TDNN and mR18 based speaker recognition systems in Experiment 1.*

	Softmax	EER [%]	minDCF	Epoch
TDNN	Standard	5,23	0,479	1,94
	AAS [13]	4,61	0,432	3,52
	Fixed Scale [17]	4,75	0,512	3,11
	SAda (AdaCos [17])	4,75	0,453	2,72
	MAda	4,70	0,439	3,11
	ParAda (a=20,b=0)	**4,29**	0,422	**1,09**
	ParAda (a=25,b=0)	4,30	0,455	1,56
	ParAda (a=20,b=0.1)	4,32	**0,418**	2,14
	ParAda (a=25,b=0.1)	4,47	0,439	1,29
mR18	Standard	4,17	0,445	2,87
	AAS [13]	3,73	0,407	2,77
	Fixed Scale [17]	4,23	0,463	2,01
	SAda (AdaCos [17])	3,93	0,446	1,68
	MAda	3,83	0,389	3,16
	ParAda a=20,b=0	3,62	0,418	1,34
	ParAda a=25,b=0	3,69	0,408	1,85
	ParAda a=20,b=0.1	3,54	**0,377**	1,67
	ParAda a=25,b=0.1	**3,49**	0,395	**1,14**

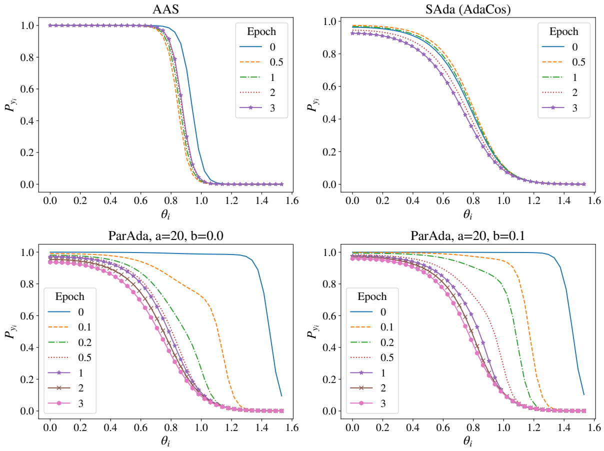


Figure 2: *Predicted probability curves $P_{y_i}(\theta_{y_i})$ at different training stages (in epochs) for TDNN system in Experiment 1.*

As evaluation metrics, we select the Equal Error Rate (*EER*) and minimum Detection Cost Function (*minDCF*) with parameters set as $C_{miss} = C_{fa} = 1$ and $P_{tar} = 0.01$. The phase of the neural network training is presented in epoch defined as one pass of all training data through the network.

4.2. Results and discussion

Table 1 presents the results of Experiment 1 for the TDNN and the proposed mR18 based speaker recognition systems. As can be clearly observed, existing modifications to the softmax-based loss function improve the EER and minDCF results, however, an increase in accuracy comes at a cost of a lower convergence speed. On the other hand, the proposed ParAda approach clearly outperforms the existing methods in terms of both the accuracy and convergence speed. In particular, the convergence speed can be increased by 2 times while the gain in EER and minDCF can also increase significantly when compared with the gains of the existing methods over the standard softmax function. Concerning the existing methods, the adaptive method known as AdaCos offers an increase in network convergence, however, its accuracy performance is lower than for the AAS method with a fixed margin value. Additional insight into the network convergence can be obtained from Fig. 2, which shows that AAS and AdaCos change the predicted probability curves only slightly during network training. In contrast, the proposed ParAda (for both considered cases) eases the network training at early training stages by shifting the probability curves to the right, and next it makes the logits more discriminative at the later stages of NN training to enhance classification ability. Less strict supervision results from the negative margin value that controls λ parameter at the initial stage. Exceeding zero value by the margin brings about stricter classification requirements.

Table 2: *Accuracy in terms of EER and minDCF, and approximate network convergence time (in Epoch) for TDNN and mR18 based speaker recognition systems in Experiment 2.*

Network	Softmax	EER [%]	minDCF	Epoch
TDNN	Standard	3,06	0,338	3,10
	AAS [13]	2,57	0,289	7,83
	AdaCos [17]	2,44	0,276	7,53
	ParAda	**2,32**	**0,257**	5,76
mR18	Standard	2,07	0,286	2,82
	AAS [13]	2,12	**0,274**	6,77
	AdaCos [17]	1,94	0,335	5,11
	ParAda	**1,72**	**0,280**	3,51

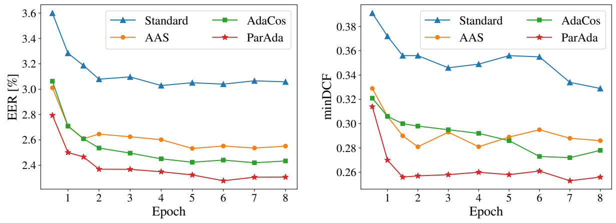


Figure 3: *EER and minDCF results at different stages of training (in epochs) for TDNN-based system in Experiment 2.*

Table 2 presents the results of Experiment 2 for four selected methods. In general, a similar trend can be observed for both network types, with ParAda clearly outperforming the other approaches in terms of accuracy and network convergence speed for the TDNN, and clearly outperforming the existing approaches in terms of the EER and convergence speed for the mR18 method (note that in case of this network, AAS achieved comparable minDCF result). For the methods studied in Experiment 2, in Fig. 3 we show speaker recognition accuracy at different stages of the network training. As can be observed, the proposed ParAda facilitates much faster network convergence at the very early part of the network training, which allows to reach high accuracy very quickly. In contrast, the compared existing approaches converge much more slowly, often not reaching the accuracy of the proposed ParAda approach.

In both experiments, the mR18 significantly outperforms the TDNN. Therefore, we can assess the proposed structure as an interesting alternative for DNN-based speaker recognition.

5. Conclusions

In this paper, we have proposed a softmax-based cross-entropy loss function with adaptive parameters (ParAda) which significantly improves speaker recognition accuracy and neural network training convergence speed compared with state-of-the-art alternatives. In addition, we have shown that the proposed modified ResNet-based architecture brings about large improvement in speaker recognition over the TDNN-based x-vector system.

6. References

[1] J. S. Chung, A. Nagrani, and A. Zisserman, “VoxCeleb2: Deep Speaker Recognition,” *Proc. Interspeech 2018*, pp. 1086–1090, 2018.

[2] 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)*. IEEE, 2018, pp. 5329–5333.

[3] M. K. Nandwana, J. van Hout, C. Richey, M. McLaren, M. A. Barrios, and A. Lawson, “The VOiCES from a Distance Challenge 2019,” in *Proc. Interspeech 2019*, 2019, pp. 2438–2442.

[4] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, “Speaker Recognition for Multi-speaker Conversations Using X-vectors,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5796–5800.

[5] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohammadi, and S. Khudanpur, “Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks,” in *Proc. Interspeech 2018*, 2018, pp. 3743–3747.

[6] S. Novoselov, A. Gusev, A. Ivanov, T. Pekhovsky, A. Shulipa, G. Lavrentyeva, V. Volokhov, and A. Kozlov, “STC Speaker Recognition Systems for the VOiCES from a Distance Challenge,” in *Proc. Interspeech 2019*, 2019, pp. 2443–2447.

[7] D. Snyder, J. Villalba, N. Chen, D. Povey, G. Sell, N. Dehak, and S. Khudanpur, “The JHU Speaker Recognition System for the VOiCES 2019 Challenge,” in *Proc. Interspeech 2019*, 2019, pp. 2468–2472.

[8] J. Huang and T. Bocklet, “Intel Far-Field Speaker Recognition System for VOiCES Challenge 2019,” in *Proc. Interspeech 2019*, 2019, pp. 2473–2477.

[9] D. Cai, X. Qin, W. Cai, and M. Li, “The DKU System for the Speaker Recognition Task of the 2019 VOiCES from a Distance Challenge,” in *Proc. Interspeech 2019*, 2019, pp. 2493–2497.

[10] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015.

[11] K. Han, D. Yu, and I. Tashev, “Speech Emotion Recognition Using Deep Neural Network and Extreme Learning Machine,” in *INTERSPEECH*, 2014, pp. 223–227.

[12] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, “SphereFace: Deep Hypersphere Embedding for Face Recognition,” *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017.

[13] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, “ArcFace: Additive Angular Margin Loss for Deep Face Recognition,” *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[14] F. Wang, J. Cheng, W. Liu, and H. Liu, “Additive Margin Softmax for Face Verification,” *IEEE Signal Processing Letters*, vol. 25, no. 7, p. 926–930, Jul 2018.

[15] Y. Liu, L. He, and J. Liu, “Large Margin Softmax Loss for Speaker Verification,” in *Proc. Interspeech 2019*, 2019, pp. 2873–2877.

[16] Y. Li, F. Gao, Z. Ou, and J. Sun, “Angular Softmax Loss for End-to-end Speaker Verification,” *2018 11th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pp. 190–194, 2018.

[17] X. Zhang, R. Zhao, Y. Qiao, X. Wang, and H. Li, “AdaCos: Adaptively Scaling Cosine Logits for Effectively Learning Deep Face Representations,” *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[18] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016.

[19] Z. Chen, Z. Ren, and S. Xu, “A Study on Angular Based Embedding Learning for Text-independent Speaker Verification,” *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Nov 2019.

[20] W. Cai, J. Chen, and M. Li, “Exploring the Encoding Layer and Loss Function in End-to-End Speaker and Language Recognition System,” *Odyssey 2018 The Speaker and Language Recognition Workshop*, Jun 2018.

[21] A. Nagrani, J. S. Chung, and A. Zisserman, “VoxCeleb: A Large-Scale Speaker Identification Dataset,” *Proc. Interspeech 2017*, pp. 2616–2620, 2017.

[22] *Room Impulse Response and Noise Database*, accessed March 9, 2020. [Online]. Available: http://www.openslr.org/28/

[23] D. Snyder, G. Chen, and D. Povey, “MUSAN: A Music, Speech, and Noise Corpus,” 2015, arXiv:1510.08484v1.

[24] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, “The Kaldi speech recognition toolkit,” IEEE Signal Processing Society, Tech. Rep., 2011.

[25] H. Zeinali, L. Burget, J. Rohdin, T. Stafylakis, and J. H. Cernocky, “How to Improve Your Speaker Embeddings Extractor in Generic Toolkits,” *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6141–6145, 2018.

II Spine2Net: SpineNet with Res2Net and Time-Squeeze-and-Excitation Blocks for Speaker Recognition

M. Rybicka, J. Villalba, P. Zelasko, N. Dehak, K. Kowalczyk "*Spine2Net: SpineNet with Res2Net and Time-Squeeze-and-Excitation Blocks for Speaker Recognition*", Interspeech, Brno, Czech Republic, 2021.

INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Spine2Net: SpineNet with Res2Net and Time-Squeeze-and-Excitation Blocks for Speaker Recognition

Magdalena Rybicka[1*], *Jesús Villalba*[2,3], *Piotr Żelasko*[2,3], *Najim Dehak*[2,3], *Konrad Kowalczyk*[1]

[1]AGH University of Science and Technology, Institute of Electronics, Kraków, Poland
[2]Center for Language and Speech Processing, Johns Hopkins University, Baltimore, USA
[3]Human Language Technology Center of Excellence, Johns Hopkins University, Baltimore, USA
{mrybicka, konrad.kowalczyk}@agh.edu.pl, {jvillal7, pzelasko, ndehak3}@jhu.edu

Abstract

Modeling speaker embeddings using deep neural networks is currently state-of-the-art in speaker recognition. Recently, ResNet-based structures have gained a broader interest, slowly becoming the baseline along with the deep-rooted Time Delay Neural Network based models. However, the scale-decreased design of the ResNet models may not preserve all of the speaker information. In this paper, we investigate the SpineNet structure with scale-permuted design to tackle this problem, in which feature size either increases or decreases depending on the processing stage in the network. Apart from the presented adjustments of the SpineNet model for the speaker recognition task, we also incorporate popular modules dedicated to the residual-like structures, namely the Res2Net and Squeeze-and-Excitation blocks, and modify them to work effectively in the presented neural network architectures. The final proposed model, i.e., the SpineNet architecture with Res2Net and Time-Squeeze-and-Excitation blocks, achieves remarkable Equal Error Rates of 0.99 and 0.92 for the Extended and Original trial lists of the well-known VoxCeleb1 dataset.

Index Terms: deep neural networks, SpineNet model, scale-permuted network, ResNet model, speaker recognition

1. Introduction

Current state-of-the-art in speaker recognition is to model speaker characteristics using deep neural networks (DNN), from which embeddings – commonly referred to as x-vectors [1] – are extracted. This approach has been shown in numerous studies [1, 2, 3] to outperform the well-established i-vector model [4]. Baseline DNN architectures include the so-called Time Delay Neural Networks (TDNN) [1], as well as their two modifications known as the Extended TDNN [5] and Factorized TDNN [6]. Recently, we have observed a rapid increase of popularity of the ResNet-based structures, with model adjustments presented e.g. in [2, 7, 8], which often offer an improved performance over the aforementioned baseline models. In [9] the authors point out that the scale-decreased design of the ResNet model may cause a removal of useful information. In order to overcome this problem, they propose a scale-permuted network design [9], where feature resolution and dimension can change arbitrarily as it is processed through the network. It outperformed i.a. ResNet with Feature Pyramid Network [10].

In this paper, we adjust the scale-permuted SpineNet structure [9] and incorporate it in the DNN-based speaker recognition model. We show that the multi-scale feature representation of SpineNet, in which input to the pooling layer is merged from several prior layers with various feature resolution, overcomes some of the limitations encountered by the scale-decreased models such as ResNet. In the context of speaker recognition, multi-scale feature resolution has been studied in [11, 12, 13, 14] for utterances of variable length, achieving notable improvement. In addition, we modify two existing residual-like structures such as Res2Net block [15] and Squeeze-and-Excitation (SE) block [16], and show that the presented modules further improve system performance. The final proposed SpineNet model with Res2Net and Time-Squeeze-and-Excitation blocks achieves highly competitive results for the trial lists of the VoxCeleb1 dataset, outperforming the known models such as [2].

2. Deep Neural Network Structures

2.1. ResNet architectures

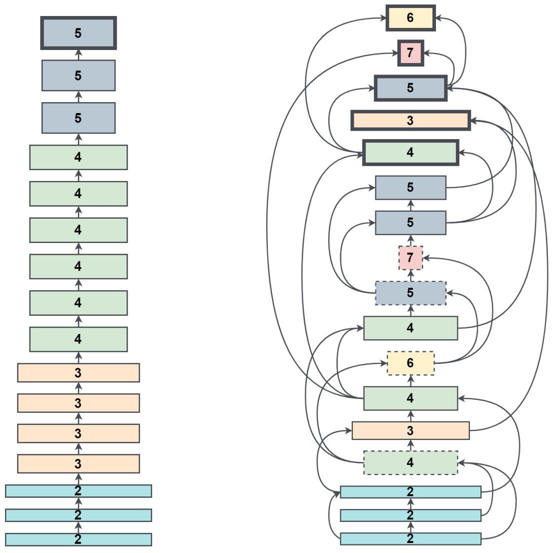
In this work, we consider three models based on the well-known ResNet architecture [17] for speaker embedding extraction. ResNet-34 and ResNet-50 [18] follow the so-called scale-decreased design, in which the size of the feature map is reduced as it is processed by the network. In both models, the sequence of the main building blocks is similar. The input feature map is passed through a 64-channel 3×3 convolutional layer with stride=1. Next, the output features are processed by the so-called residual part of the structure, which is presented in Figure 1a for ResNet-50. The residual part of the ResNet-34 exhibits the same block sequence, with the difference that the bottleneck residual blocks are replaced with the basic residual blocks [18]. As shown in Figure 1a, residual blocks are distributed across several levels (2-5). At the beginning of each level–starting at level 3– the feature maps are downsampled by 2. Thus, a block at level l has feature maps downsampled by a factor 2^{l-2}. Table 1 presents the resulting feature map size and the number of base channels for the blocks at each level. Note that for the basic residual block, the number of output channels is equal to the number of the base channels, while the bottleneck block increases the output channels by 4. In both networks, the output of the residual part is passed to the statistics pooling, followed by the fully connected layer along with the softmax layer. In our experiments, we also considered a light version of the ResNet-34 model, in which the number of channels of all blocks were decreased by 4. Hereafter, we will refer to this structure as Thin-ResNet-34.

2.2. SpineNet architectures

SpineNet structure belongs to the family of scale-permuted meta-architectures [9]. The inner connections, block order and

[*]work performed while at Center for Language and Speech Processing, Johns Hopkins University.

http://dx.doi.org/10.21437/Interspeech.2021-1163



(a) *ResNet-50*

(b) *SpineNet-49*

Figure 1: *The structure of the residual part of (a) ResNet-50 and (b) SpineNet-49 networks. In diagrams, bottleneck blocks are marked with solid lines, basic residual blocks are marked with dotted lines. Blocks with bold line contours (located in the top of the architectures) represent the output blocks, while the number inside each block indicates its level.*

Table 1: *Sizes of features for blocks corresponding to the levels* $l = 2, 3, 4, 5, 6,$ *and* 7 *for the ResNet and SpineNet models. The feature map size is given in terms of F (feature) and T (time) lengths, while the feature dimension is given by the number of base channels.*

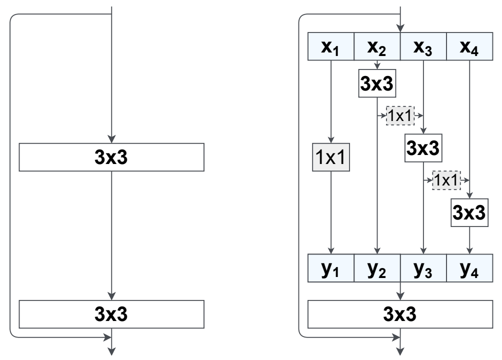
Block level	Feature map size	No. base channels	
		ResNet	SpineNet
2	F × T	64	64
3	F/2 × T/2	128	128
4	F/4 × T/4	256	256
5	F/8 × T/8	512	256
6	F/16 × T/16	—	256
7	F/32 × T/32	—	256

their type is derived by Neural Architecture Search in [9], with ResNet-50 incorporated as a baseline.

Similar to the ResNet model, the network input is first processed by the 3×3 convolutional layer with stride of 1×1. The next part of the structure is presented in Figure 1b. It consists of stem scale-decreased and learned scale-permuted segments.

The stem network part is represented by the first two bottleneck blocks at the second level, whose outputs are used as candidate input features for the scale-permuted segment. In the scale-decreased network, block sequence follows a fixed order, where block level is kept unchanged or it is increased with the network processing flow.

The scale-permuted part is build of blocks that arbitrarily increase or decrease its level with the network processing flow. In this segment, blocks accept 2 input connections, where output blocks, indicated by bold contour lines in Figure 1b, accept up to 3 inputs. Input features are fused by an element-wise addition. Since the connections between the blocks are cross-scale, each connection consists of 3 components: (i) 1×1 convolution, which reduces the number of channels by a factor $\alpha = 0.5$ compared with the number of base channels of the block from which the connection is made, (ii) feature map resampling operation; and (iii) another 1×1 convolution which transforms the channel number to the target size. The target number of channels for the basic residual blocks is equal to the number of base channels, while the target number of channels for the bottleneck blocks is 4 times larger. In contrast to the ResNet structures, SpineNet incorporates both types of residual blocks. Table 1 presents the feature map sizes and the number of base channels associated with each level of the structure. In the resampling part, the upsampling operation is performed by the nearest-neighbor interpolation. The downsampling is achieved by the convolution of size 3×3, with stride 2. If necessary the convolution is followed by maximum pooling with a 3×3 kernel and a stride of 2 or alternatively with a 5×5 kernel and a stride of 4.



(a) *Basic residual block*

(b) *Basic Res2Net block*

Figure 2: *Basic residual block and its proposed Res2Net adaptation. Example presented for the scale* $s = 4$.

Output block features (marked with bold contour lines in the Figure 1b) are processed by 1×1 endpoint convolutions to obtain the common number of channels, which we set to 256. Next, the feature maps are upsampled with nearest-neighbour interpolation to match the feature map size at the lowest level. Representations from the different levels are merged with a point-wise average operation and are forwarded to the statistics pooling followed by the fully connected and softmax layers.

In this work, we also present the results for the two modifications of the SpineNet-49 structure, namely the so-called Thin-SpineNet-49 and SpineNet-49S. The former follows similar modification as in ResNet, i.e. the number of channels of all layers is reduced 4 times (scaling factor of 0.25), including the number of the endpoint filters. The latter model, SpineNet-49S, represents the intermediate structure between the Thin-SpineNet-49 and SpineNet-49, where filter dimensions are decreased using a scaling factor of 0.66 (except for the input convolution) and the number of endpoint channels is set to 128. In the original paper [9], the SpineNet-49S has an associated factor of 0.65, which we modify in order to obtain the desired number of channels such that the modifications described in the next two subsections were feasible.

2.3. Res2Net module

In this work, we incorporate the so-called Res2Net modules [15] into ResNet and SpineNet. Res2Net introduces a new dimension - scale s, which increases the receptive field and granular level of the bottleneck residual block.

Res2Net blocks were proposed as substitute for the residual bottleneck blocks in structures like ResNet-50 and larger. Since SpineNet also includes basic residual blocks, we adapted the original Res2Net structure to the basic block. Our modification is presented in Figure 2. Input of the basic Res2Net block is

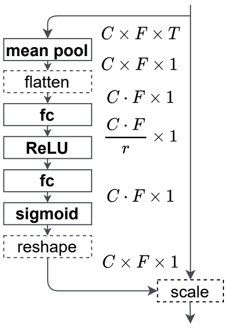


Figure 3: *Diagram of the Time-Squeeze-and-Excitation (T-SE) module, where C, F, and T denote the number of input channels, the frequency dimension, and the time dimension, while r is the reduction ratio.*

split evenly into s parts along the channel dimension. Then, each x_i group, where $i = 1, ..., s$, is processed by a separate convolutional layer with w output channels. The corresponding output y_i can be expressed as follows:

$$y_i = \begin{cases} \mathrm{K}_{1\times1}(x_i) & \text{if } i = 1 \\ \mathrm{K}_{3\times3}(x_i) & \text{if } i = 2 \\ \mathrm{K}_{3\times3}(x_i + y_{i-1}) & \text{if } s \geq i > 2, w = C_{\text{in}}/s \\ \mathrm{K}_{3\times3}(x_i + \mathrm{K}_{1\times1}(y_{i-1})) & \text{if } s \geq i > 2, w \neq C_{\text{in}}/s \end{cases} \quad (1)$$

where $\mathrm{K}_{1\times1}$ and $\mathrm{K}_{3\times3}$ are convolutional layers with a 1×1 and 3×3 kernels respectively; and C_{in} are the block input channels. The inner convolution channels w can be set equal to the number of input channels, or alternatively it can be increased to preserve/extend network complexity. For the latter case, we need projection convolutions $\mathrm{K}_{1\times1}$ in the inner residual connections to match the channel dimensions and thereby enable the addition operation. Concatenated y_i features are passed to another convolutional layer (kernel of 3×3). In this work, we incorporate the scale $s = 4$ and we set $w = 26$ for the inner convolutions for residual blocks with 64 base channels (level 2). This increased the total number of inner channels from 64 to 104 w.r.t. the standard residual blocks. We observed that such an increase in the Res2Net was required to maintain the performance. The value of w for the blocks with a higher number of base channels is increased proportionally to the block size. We will refer to the SpineNet with Res2Net blocks as Spine2Net.

2.4. (Time-)Squeeze-and-Excitation blocks

Squeeze-and-Excitation (SE) blocks [16] constitute a common approach to re-calibrate channel dependencies. In case of speaker recognition, the frequency dependencies are also of high importance. Therefore, we enhance the modelling capabilities of SE blocks by introducing the Time-Squeeze-and-Excitation (T-SE) module [19]. The T-SE model is similar to the SE model, however, the average pooling is applied only along the time dimension, instead of the global pooling of the entire feature map. The algorithm pipeline is presented in Figure 3. The squeeze operation produces the channel- and frequency-wise descriptor by applying mean pooling along the time axis. This process is followed by an excitation step, in which calibration weights are estimated. These weights are computed by a dimensionality reduction layer with reduction factor r, ReLU non-linearity, and fully-connected layer with sigmoid activation. Obtained scale values are then used to re-calibrate the feature maps. Note that, original SE pools over time and frequency axis producing only a single scale value per channel. Thus, all frequency bins are scaled by the same value, while T-SE produces a different scaling per bin.

3. Experimental Evaluation and Results

3.1. Datasets, system framework and evaluation measures

This section presents the datasets and general framework of the evaluated speaker recognition systems. Neural networks were trained on VoxCeleb2 [20] with 6112 speakers. Utterances derived from the same video were concatenated and extended with 3 types of noise augmentations: music, environmental noise, babble speech from the MUSAN corpus [21], and reverberation with three sets of room impulse responses (RIRs) [22]. The test set is based on the VoxCeleb1 corpus [23] and clean versions of trials: Extended (VoxCeleb1-E), Hard (VoxCeleb1-H), and Original (VoxCeleb1-O) [24]. One epoch of the training consisted of randomly selected, augmented 4 s chunks in the number equal to the number of all augmented training utterances. Each network was trained for 70 epochs.

Input features were 80-dimensional log-Mel filter-banks extracted from 25 ms sliding window with 10 ms shift, and mean-normalization over a 3 s window. Speech frames were selected with Kaldi energy-based Voice Activity Detector [25]. The neural network was trained with Additive Angular Softmax [26] loss with scale $s_{\text{AAS}} = 30$ and margin $m_{\text{AAS}} = 0.3$. The margin was linearly increased from 0 to 0.3 during the first 20 epochs. Speaker embeddings were extracted from the penultimate fully connected layer, which yields a feature vector of length 256. In the system backend, we used cosine scoring as it provided better results than PLDA. The network structure was implemented in PyTorch [27] and it was trained with Adam optimizer [28] along with an exponential learning rate scheduler with an initial value of 0.05 [29, 30]. For experiments incorporating the SE blocks, the reduction factor was set to $r_{\text{SE}} = 16$. T-SE blocks had a larger value of $r_{\text{T-SE}} = 256$, as the T-SE blocks significantly enlarge the network size.

As evaluation measures, we used the Equal Error Rate (EER), reported in %, and minimum Detection Cost Function [31] with $P_{tar} = 0.05$ (DCF5) and $P_{tar} = 0.01$ (DCF1). The FLOP values were calculated for a 3 s utterance, with multiply-add counted as a single operation.

3.2. Results and discussion

Table 2 presents the results of a preliminary comparison of Thin versions of ResNet-34 and SpineNet-49. First, we compare two structures based only on single-scale features (lines 1-2). The output from the last layer in Thin-ResNet-34 and the output from the last block at the 5th level from the Thin-SpineNet-49 (denoted as Thin-SpineNet-49-5) were taken as single feature representations for pooling input. For fair comparison, Thin-SpineNet-49-5 representations were directly forwarded to the next layers without endpoint post-processing. Both structures provided similar performance, with a slight predominance of the ResNet model. The second experiment (lines 3-4) presents the gain offered by using multi-scale features. Thin-ResNet-34 modified to produce multi-scale features is denoted as Thin-ResNet-34-345. To this end, we incorporated the features from the last layer of blocks at the 3, 4, and 5th levels. As for the SpineNet models, the features were processed with a 1×1 convolutional layer transforming the channel number to 64 and upsampling the feature maps to the size of the 3rd level, followed by an average across levels. We observe that, for ResNet, multi-scale feature fusion did not significantly improve, except for

Table 2: *Results of experimental evaluation of Thin SpineNet and ResNet structures on the VoxCeleb1 test datasets.*

			VoxCeleb1-E			VoxCeleb1-H			VoxCeleb1-O		
Network	# Params	# FLOPs	EER	DCF5	DCF1	EER	DCF5	DCF1	EER	DCF5	DCF1
Thin-ResNet-34	3.6M	1.7G	1.90	0.119	0.200	3.25	0.189	0.298	2.05	0.149	0.240
Thin-SpineNet-49-5	3.9M	1.4G	1.95	0.123	0.209	3.28	0.189	0.298	2.07	0.135	0.211
Thin-ResNet-34-345	4.2M	1.7G	1.89	0.120	0.208	3.27	0.191	0.299	1.99	0.135	0.217
Thin-SpineNet-49	4.3M	1.7G	1.83	0.117	0.196	3.20	0.184	0.293	1.84	0.127	0.209

Table 3: *Results of experimental evaluation of SpineNet and ResNet structures, along with introduced modifications including Res2Net modules, Squeeze-and-Excitation (SE) blocks, and the Time-Squeeze-and-Excitation (T-SE) blocks on the VoxCeleb1 test datasets.*

			VoxCeleb1-E			VoxCeleb1-H			VoxCeleb1-O		
Network	# Params	# FLOPs	EER	DCF5	DCF1	EER	DCF5	DCF1	EER	DCF5	DCF1
ResNet-34	25.5M	27.3G	1.19	0.078	0.140	2.27	0.137	0.219	1.35	0.088	0.146
ResNet-50	35.6M	30.7G	1.30	0.082	0.150	2.33	0.142	0.235	1.44	0.100	0.173
SpineNet-49S	13.5M	11.2G	1.25	0.079	0.138	2.29	0.137	0.226	1.11	0.069	0.120
SpineNet-49	28.6M	26.0G	1.17	0.074	0.129	2.14	0.129	0.213	1.11	0.088	0.125
Res2Net-34	26.1M	27.6G	1.16	0.074	0.130	2.17	0.128	0.218	1.18	0.078	0.115
Res2Net-50	35.7M	32.0G	1.09	0.068	0.122	2.00	0.119	0.195	1.14	0.081	0.116
Spine2Net-49S	13.5M	11.3G	1.13	0.073	0.131	2.18	0.130	0.210	1.02	0.077	0.137
Spine2Net-49	28.8M	26.2G	1.10	0.071	0.127	2.18	0.132	0.216	1.09	0.070	0.116
SE-Res2Net-50	38.2M	32.0G	1.24	0.080	0.140	2.45	0.141	0.225	1.31	0.084	0.132
SE-Spine2Net-49S	14.0M	11.3G	1.09	0.069	0.122	2.11	0.124	0.205	1.05	0.066	0.104
SE-Spine2Net-49	29.8M	26.2G	1.04	0.067	0.119	2.07	0.121	0.206	1.14	0.066	0.098
T-SE-Res2Net-50	88.1M	32.1G	1.05	0.067	0.117	1.95	0.113	0.196	1.12	0.071	0.103
T-SE-Spine2Net-49S	26.0M	11.3G	1.08	0.070	0.124	2.09	0.124	0.204	1.08	0.074	0.127
T-SE-Spine2Net-49	58.0M	26.2G	0.99	0.065	0.112	1.95	0.117	0.192	0.92	0.068	0.105

VoxCeleb1-O. On the other hand, the Thin-SpineNet-49 benefited from the multi-scale representations, outperforming the other structures in all three test datasets.

Table 3 presents the results of a set of experiments of architectures with large, more complex structures and channel numbers as in their original form. We report on the results of four experiments: the baseline results (original structures), additional incorporation of the Res2Net modules, incorporation of the Squeeze-and-Excitation (SE) blocks on top of the previous alterations, and incorporation of the presented Time-Squeeze-and-Excitation (T-SE) blocks instead of SE blocks.

The first block of the table compares four basic structures, namely ResNet-34, ResNet-50, SpineNet-49S, and SpineNet-49. Comparing ResNets, ResNet-50 did not provide any gain over ResNet-34, despite having more learnable parameters. SpineNet-49S improved over ResNet-50 and presented competitive results to the ResNet-34 model. Note that SpineNet-49S has half of parameters and FLOPs than ResNet-34. SpineNet-49 clearly outperformed both ResNet structures for all the test datasets. Furthermore, the SpineNet-49 structure provided better performance with a lower number of FLOPs than ResNet-34, although it has more parameters. It is important to note that the reported number of parameters and FLOPs does not imply directly an improved training speed of the network but rather the structure effectiveness with respect to the model size.

The second block of the table introduces the Res2Net blocks to the described structures. In all four models, this modification provided a notable gain for speaker recognition measures–except the Hard condition for the Spine2Net-49, which achieved a comparable result than its original model. The Res2Net-34 model presents the effectiveness of the proposed Res2Net module adaptation for the basic residual block, reporting a clear gain over ResNet-34. Res2Net-50 achieved significantly better results than the ResNet-50 with 21% and 33% of relative improvement for EER and DCF1 for the VoxCeleb1-O dataset. Similar as in the previous evaluation, SpineNet-49S appears to be a competitive structure, outperforming the Res2Net-34 in nearly all test scenarios, while having less than half of the computational cost. Since Res2Net-50 presents clearly better results than Res2Net-34, in the following experiments we focus on the Res2Net-50 model only. In the third block of the table, we incorporate SE modules to the previous structures. The reported results show evidently accuracy degradation for the SE-Res2Net-50, whereas both Spine2Net structures clearly benefit from introduction of the re-calibration module. In final evaluation, the SE modules are replaced with the T-SE modules. The gain achieved by this replacement is evident. The least relative improvement can be observed for the T-SE-Spine2Net-49S architecture, nevertheless, the model still performs competitive, offering high recognition accuracy and low complexity. Note that the T-SE block strongly increases the number of network parameters, whereas the number of FLOPs is kept almost intact. Among all compared models, the last proposed structure, namely the T-SE-Spine2Net-49 model, provides an outstanding performance, outperforming all other systems.

4. Conclusions

This paper investigated the application of the scale-permuted architecture known as SpineNet for the speaker recognition task. Having adjusting the model, we incorporated Res2Net and Squeeze-and-Exciation modules, and proposed their modifications to achieve superb performance in the studied task. The results of experiments demonstrate that speaker recognition accuracy benefits from adopting the SpineNet structure and its multi-scale feature representation. Furthermore, the proposed Res2Net and T-SE modules further boost its performance.

5. Acknowledgements

This research was supported by the Foundation for Polish Science under grant number First TEAM/2017-3/23 (POIR.04.04.00-00-3FC4/17-00) which is co-financed by the European Union, and in part by PLGrid Infrastructure.

6. References

[1] 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)*, 2018, pp. 5329–5333.

[2] J. Villalba, N. Chen, D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, J. Borgstrom, L. P. García-Perera, F. Richardson, R. Dehak, P. A. Torres-Carrasquillo, and N. Dehak, "State-of-the-art speaker recognition with neural network embeddings in NIST SRE18 and Speakers in the Wild evaluations," *Computer Speech & Language*, vol. 60, p. 101026, 2020.

[3] M. K. Nandwana, J. van Hout, M. McLaren, A. Stauffer, C. Richey, A. Lawson, and M. Graciarena, "Robust Speaker Recognition from Distant Speech under Real Reverberant Environments Using Speaker Embeddings," in *Proc. Interspeech 2018*, 2018, pp. 1106–1110.

[4] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-End Factor Analysis for Speaker Verification," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 4, pp. 788–798, 2010.

[5] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, "Speaker Recognition for Multi-speaker Conversations Using X-vectors," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5796–5800.

[6] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohammadi, and S. Khudanpur, "Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks," in *Proc. Interspeech 2018*, 2018, pp. 3743–3747.

[7] M. Rybicka and K. Kowalczyk, "On Parameter Adaptation in Softmax-Based Cross-Entropy Loss for Improved Convergence Speed and Accuracy in DNN-Based Speaker Recognition," in *Proc. Interspeech 2020*, 2020, pp. 3805–3809.

[8] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification," in *Proc. Interspeech 2020*, 2020, pp. 3830–3834.

[9] X. Du, T. Y. Lin, P. Jin, G. Ghiasi, M. Tan, Y. Cui, Q. V. Le, and X. Song, "SpineNet: Learning Scale-Permuted Backbone for Recognition and Localization," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 589–11 598.

[10] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 936–944, 2017.

[11] Y. Jung, S. M. Kye, Y. Choi, M. Jung, and H. Kim, "Improving Multi-Scale Aggregation Using Feature Pyramid Module for Robust Speaker Verification of Variable-Duration Utterances," in *Proc. Interspeech 2020*, 2020, pp. 1501–1505.

[12] Z. Gao, Y. Song, I. McLoughlin, P. Li, Y. Jiang, and L.-R. Dai, "Improving Aggregation and Loss Function for Better Embedding Learning in End-to-End Speaker Verification System," in *Proc. Interspeech 2019*, 2019, pp. 361–365.

[13] S. Seo, D. J. Rim, M. Lim, D. Lee, H. Park, J. Oh, C. Kim, and J.-H. Kim, "Shortcut Connections Based Deep Speaker Embeddings for End-to-End Speaker Verification System," in *Proc. Interspeech 2019*, 2019, pp. 2928–2932.

[14] A. Hajavi and A. Etemad, "A Deep Neural Network for Short-Segment Speaker Recognition," in *Proc. Interspeech 2019*, 2019, pp. 2878–2882.

[15] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2Net: A New Multi-Scale Backbone Architecture," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 2, p. 652–662, Feb 2021.

[16] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.

[17] H. Zeinali, S. Wang, A. Silnova, P. Matejka, and O. Plchot, "BUT System Description to VoxCeleb Speaker Recognition Challenge 2019," *arXiv preprint arXiv:1910.12592*, 2019.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Jun 2016.

[19] S. Kataria, P. S. Nidadavolu, J. Villalba, N. Chen, P. Garcia-Perera, and N. Dehak, "Feature Enhancement with Deep Feature Losses for Speaker Verification," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7584–7588.

[20] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep Speaker Recognition," in *Proc. Interspeech 2018*, 2018, pp. 1086–1090.

[21] D. Snyder, G. Chen, and D. Povey, "MUSAN: A Music, Speech, and Noise Corpus," 2015, arXiv:1510.08484v1.

[22] *Room Impulse Response and Noise Database*, accessed February 17, 2021. [Online]. Available: http://www.openslr.org/28/

[23] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A Large-Scale Speaker Identification Dataset," *Proc. Interspeech 2017*, pp. 2616–2620, 2017.

[24] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Computer Speech & Language*, vol. 60, p. 101027, 2020.

[25] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi Speech Recognition Toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, Dec. 2011.

[26] 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)*, 2019, pp. 4685–4694.

[27] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035.

[28] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[29] J. A. Villalba Lopez, D. Garcia-Romero, N. Chen, G. Sell, J. Borgstrom, A. McCree, L. P. Garcia Perera, S. Kataria, P. S. Nidadavolu, P. Torres-Carrasquiilo, and N. Dehak, "Advances in Speaker Recognition for Telephone and Audio-Visual Data: the JHU-MIT Submission for NIST SRE19," in *Proc. Odyssey 2020 The Speaker and Language Recognition Workshop*, 2020, pp. 273–280.

[30] J. Villalba and N. Dehak, *The JHU System Description for SDSV2020 Challenge*, accessed June 10, 2021. [Online]. Available: https://sdsvc.github.io/2020/descriptions/Team10_Both.pdf

[31] NIST, *NIST 2018 Speaker Recognition Evaluation Plan*, accessed June 10, 2021. [Online]. Available: https://www.nist.gov/system/files/documents/2018/08/17/sre18_eval_plan_2018-05-31_v6.pdf

III End-to-End Neural Speaker Diarization with an Iterative Refinement of Non-Autoregressive Attention-based Attractors

M. Rybicka, J. Villalba, N. Dehak and K. Kowalczyk, "*End-to-End Neural Speaker Diarization with an Iterative Refinement of Non-Autoregressive Attention-based Attractors*", Interspeech, Incheon, South Korea, 2022.

Interspeech 2022
18-22 September 2022, Incheon, Korea

End-to-End Neural Speaker Diarization with an Iterative Refinement of Non-Autoregressive Attention-based Attractors

Magdalena Rybicka[1], *Jesús Villalba*[2,3], *Najim Dehak*[2,3], *Konrad Kowalczyk*[1]

[1]AGH University of Science and Technology, Institute of Electronics, Kraków, Poland
[2]Center for Language and Speech Processing, Johns Hopkins University, Baltimore, USA
[3]Human Language Technology Center of Excellence, Johns Hopkins University, Baltimore, USA
{mrybicka, konrad.kowalczyk}@agh.edu.pl, {jvillal7, ndehak3}@jhu.edu

Abstract

End-to-end neural speaker diarization (EEND) systems are currently of high interest as the approach can easily handle overlapped speech and can be trained to optimize directly the diarization decision. Recently, there have been several investigations that achieve further enhancement of the EEND system, such as proposing various network structures for the encoder module or integration of the EEND with, the well-established in speaker embedding-based diarization, clustering methods. In this paper, we propose an alternative for the EEND back-end and replace the LSTM-based attractor estimator with a non-autoregressive approach based on a Transformer decoder. Moreover, we introduce an iterative method that refines the system decision and the attractors in turns. Finally, we present results derived from an additional regularization of the proposed system with the use of Additive Angular Softmax speaker classification loss. We achieve up to 15% relative improvement over baseline on 2-speaker real recordings from CALLHOME dataset and up to 18% on simulated 2-speaker mixtures.

Index Terms: speaker diarization, end-to-end, clustering, self-attention, attractor mechanism, iterative refinement

1. Introduction

The well-established approach of handling the diarization problem are cluster-based methods [1, 2]. In general, the processing flow is based on extracting a sequence of speaker embeddings from overlapping audio segments, scoring one with each other and applying a clustering algorithm on top of those scores. The process can be preceded with Voice Activity Detection (VAD) in order to remove non-speech frames. In this approach, each of the modules is independent and is optimized separately, instead of being optimized to solve the diarization problem. Another major drawback is that speech overlap is not addressed properly, as clustering methods assume that each segment belongs to a single speaker only.

The aforementioned problems were addressed by the recently proposed end-to-end diarization (EEND) systems. The EEND has been firstly proposed as a simple multi-label classification task [3] in order to replace the clustering-based methods. The method has been improved with self-attention structured encoder [4, 5] and encoder-decoder attractor (EDA) mechanism [6, 7] that can handle a flexible number of speakers.

There has been several extensions to further improve the proposed framework. In [8, 9], the authors replace the Transformer encoder with a Conformer model. In [10, 11, 12], an intermediate approach has been proposed, in which the EEND system is combined with clustering in order to leverage the advantages offered by both methods. Also in [11], the authors show the advantage of adding an additional speaker classification loss to the training objective.

In this paper, based on the EEND-EDA system, we propose to replace the EDA module with the non-autoregressive attractor generation in which all outputs are computed independently, in parallel. The idea is inspired from Automatic Speech Recognition (ASR), in which the introduction of a non-autoregressive system has led not only to a great speed-up in computations, but also improvement in the accuracy of ASR [13, 14, 15, 16]. Moreover, in order to deal with the limitations of the proposed system, we apply an iterative refinement of the diarization outputs. A similar approach has also been used in the sequence modeling and ASR task [17, 18].

The main contributions of this paper are as follows. We propose the non-autoregressive back-end for the estimation of the attractors for speaker diarization as a new approach for attractor estimation. Next, we show that introducing iterative refinement for diarization can boost system accuracy. Finally, we present the results when incorporating the speaker classification loss as an additional regularizer for the diarization system.

2. End-to-end neural speaker diarization

In this section, we present an overview of the self-attentive end-to-end diarization model with encoder-decoder based attractors (EEND-EDA) [6]. We distinguish two main modules in the framework, namely the EEND encoder and the EDA mechanism which is considered as the diarization system back-end.

As input, the EEND receives the T-length feature sequence, denoted as $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$. At the output, the encoder produces the T-length sequence of embeddings $\mathbf{e} = \{\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_T\}$, where each embedding corresponds to a single input feature. The encoder structure is composed of 4 stacked Transformer encoder layers with self-attention mechanism. Similarly to [4], the positional encoding is omitted in the encoder structure.

The obtained embedding representations are forwarded to the back-end, namely to the EDA module. The EDA produces attractors with the number equal to the total number of speakers occurring in a particular utterance. The EDA module is built of two LSTM layers connected in the encoder-decoder manner. In theory, the EDA module can produce an infinite number of attractors. In order to constrain and distinguish whether the produced attractor represents a subsequent speaker track, a linear layer followed by a sigmoid activation is located at the top of the EDA module. The output is used to assess whether a particular embedding is useful or the attractor generation should be terminated.

In the last step, the estimated attractors are used to compute

10.21437/Interspeech.2022-10169

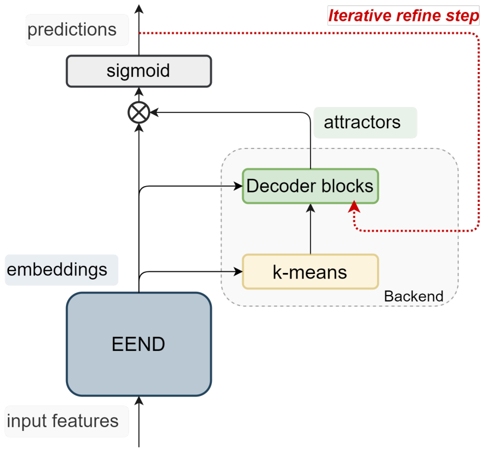


Figure 1: *Flow diagram of the EEND with the proposed non-autoregressive back-end. In the first iteration, the k-means initialization is used. In the next iterations, the output from the previous iterative refinement step is used instead.*

a dot product with each embedding separately. The obtained score is processed by a sigmoid function and constitutes the final diarization result.

The training objective is a sum of two losses, namely the attractor and the diarization loss, whereby both of them are based on the binary cross-entropy. The attractor loss is derived from the decisions of the binary classification layer located at the top of the EDA module. As ground truth, $S + 1$ binary labels are used, where S indicates the number of speakers in the recording. In order to indicate the stoppage of attractor generation, S first labels need to be set to 1 and the $S + 1$-th label needs to be set to 0. The labels are compared with the values obtained from the binary classifier at the top of the EDA module. In turn, the diarization loss is computed by comparing the diarization decisions with binary ground truth labels $\mathbf{y} \in \{0, 1\}^{T \times S}$, where $y_{t,s} = 1$ if speaker s is present at time t. The final diarization prediction is selected based on the permutation invariant training (PIT) [19] scheme. For more details on computation of both loss functions, the reader is referred to [6].

3. Proposed approach

3.1. Non-autoregressive attractor estimation

In this section, let us describe the framework of the proposed non-autoregressive back-end for the attractor estimation. In Figure 1, we present the flow diagram of the proposed system. Similar to the baseline, the input features are first processed by the EEND encoder, which is built of four-layer Transformer encoder, producing frame-level embedding representations. Such embeddings are used in the back-end in order to estimate attractor representations. The initial values of attractor representations are produced using the k-means algorithm, which clusters the embeddings of a recording. The number of clusters is equal to the number of speakers in the recording. The calculated cluster centers are taken as initial attractor values, denoted hereafter as $\mathbf{c}_s = \{\mathbf{c}_1, .., \mathbf{c}_S\}$, which in turn are forwarded to the decoder blocks that refine these representations.

A single decoder block is represented by a Transformer decoder layer [20]. The diagram of the back-end along with the detailed structure of the N-th decoder block are presented in Figure 2. A single Transformer decoder block is composed of 3 core layers: multi-head self-attention layer, multi-head source-target attention layer, and position-wise feed-forward layer (FF). Note that in Figure 2, both attention layers are denoted as MHA blocks. All of the layers are followed by the residual connection, which adds up layer input and output, and a normalization layer. The core elements in the attention mechanism are the key, value, and query linear transformations. The difference between the mentioned self-attention and source-target is in the input to these elements. Self-attention operates only on the estimated cluster centers or input from the previous decoder block. In contrast, the source-target attention incorporates the embedding representations to compute the key and value, whereby intermediate attractor estimations are used as query. Similar as in the encoder part, we do not incorporate the positional encoding. The length normalization of the vectors is applied on the embeddings before the decoder.

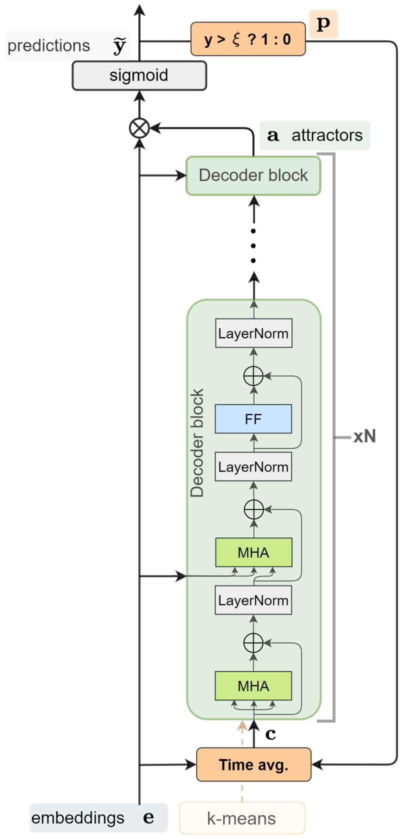


Figure 2: *Backend architecture of the proposed system with iterative refinement scheme, showing the detailed structure of one of* N *subsequently repeated decoder blocks.*

Similarly to the baseline approach, the attractors are used to compute the dot product with the embeddings, which is followed by a sigmoid function, resulting in the final diarization result. The training objective of the system is composed only of the diarization loss with the PIT scheme. In sections to follow, we will refer to the proposed system as EEND-NAA (EEND with Non-Autoregressive Attractor).

3.2. Iterative refinement

The aim of using k-means clustering at the backend is to give the first, rough estimates of the attractors, which are subsequently refined by the next layers. Such an initialization has a few drawbacks. Firstly, it operates on all embeddings extracted from the recording, and hence it can cluster not only the speaker embeddings but also the silence embeddings. Secondly, it does not take into consideration whether the particular embedding belongs to more than one cluster, such as is the case for embeddings produced for overlapped speech.

As a remedy to these issues, we incorporate the procedure of an iterative refinement of the attractors. After the first estimation of the attractors, which is computed with initialization from k-means algorithm, the diarization result is obtained $\widetilde{y}_{t,s}$. It is produced through the computation of the dot product of the attractors and encoder embeddings, followed by the sigmoid function, i.e. as

$$\widetilde{y}^{\,i}_{t,s} = \sigma(\mathbf{e}_t \otimes \mathbf{a}^i_s) \ , \tag{1}$$

where i denotes the iteration number, t is the time step, s is the speaker track, and $\mathbf{a}$ is the attractor representation. In order to obtain the estimated binary diarization decision, for each diarization decision track, we compare the obtained result with the neutral threshold value $\xi = 0.5$, which can be written as

$$p^i_{t,s} = \begin{cases} 1 & \widetilde{y}^{\,i}_{t,s} > \xi \\ 0 & \widetilde{y}^{\,i}_{t,s} \leq \xi \end{cases} . \tag{2}$$

Based on this result, we apply the refinement step where we compute new initial cluster centers, which are forward-passed through the decoder stack to estimate refined attractors. Thus from the second iteration on-wards, the initial representations are recomputed in the following manner:

$$\mathbf{c}^i_s = \frac{1}{\sum_{t=1}^{t=T} p^{i-1}_{t,s}} \sum_{t=1}^{t=T} p^{i-1}_{t,s} \cdot \mathbf{e}_t \ . \tag{3}$$

By recalculating the cluster centers, the embeddings $\mathbf{e}_t$ that represents overlap speech are incorporated into the computation of the centers for all speakers that occur at time t. Note that the aforementioned procedure applies to all iterations with $i > 1$, while for the first iteration $i = 1$, the initial representations $\mathbf{c}$ are calculated by k-means clustering.

3.3. Additional speaker classification loss

Since the representations on the encoder output can provide relative speaker information, we examine whether adding the speaker classification loss as an additional regularization at its output can further improve the embedding discrimination.

Firstly, the speaker embeddings with respect to the whole sequence are estimated based on the encoder embeddings, with the same weighted pooling method as in equation (3) and $i = I$, where I denotes the index of the final iteration. Then, the representations are fed to the speaker classification layer, with the number of outputs equal to the total number of speakers that occur in the training dataset. As a speaker loss $\mathcal{L}_{\text{spk}}$ we incorporate the Additive Angular Softmax loss function [21]. The assignment of the estimated speaker embedding to the specific speaker is based on the PIT result from diarization.

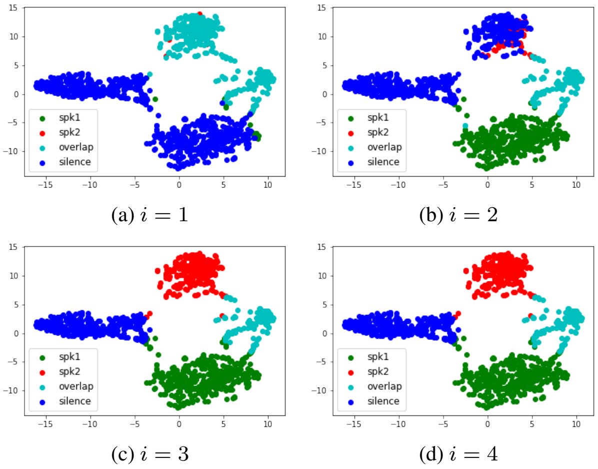
The final loss function $\mathcal{L}$ is calculated as a weighted sum of the diarization loss $\mathcal{L}_{\text{diar}}$ and the speaker classification loss $\mathcal{L}_{\text{spk}}$ in the following manner:

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_{\text{diar}} + \lambda\mathcal{L}_{\text{spk}}, \tag{4}$$

which is controlled by empirically selected λ parameter.

Table 1: *The statistics of datasets used in evaluations, including Sim2Spk sets for parameter $\beta = 1, 2, 3, 5$. All values are in %.*

Dataset	Sim2Spk				CH
	1	2	3	5	
overlap / speech	47.32	35.32	28.00	19.64	13.04
overlap / total	42.50	27.98	19.70	11.14	11.76
speech / total	89.82	79.23	70.37	56.72	90.15



(a) $i = 1$

(b) $i = 2$

(c) $i = 3$

(d) $i = 4$

Figure 3: *Visualization of the encoder embeddings obtained by EEND-NAA (with $I = 4$) at each i-th refinement iteration for an example utterance. As labels, system decisions were used.*

4. Experimental evaluation

4.1. Datasets

The experimental scenario is analogous to the one presented in [6]. Following the procedure presented in [4], we created the simulated 2-speaker mixtures (Sim2Spk) based on the following datasets: Switchboard-2 (Phase I, II, III), Switchboard Cellular (Part 1 and 2), and the NIST Speaker Recognition Evaluation (2004, 2005, 2006, 2008). MUSAN [22] dataset was used for noise augmentation, while [23] was used for reverberation. The training set was composed of 100 000 mixtures, with 35.3% speech overlap, which was accomplished by setting parameter β from [4], which controls the amount of silence in a simulated utterance, to 2. We also created four test sets consisting of 500 mixtures of 2 speakers each, for different speech overlap of 47.3%, 35.3%, 28.0%, 19.6% (by setting $\beta = 1, 2, 3, 5$).

In addition, the evaluation protocol incorporated also real, 2-speaker recordings from the CALLHOME (CH) corpus [24]. The subset was splitted into the training part used for network fine-tuning, and the test part utilized for evaluation. The split has been done as described in [4], which resulted in 155 recordings for the adaptation set and 148 recordings for the test set. The statistics of the duration of speech, overlap with respect to the total recording duration for all test datasets is presented in Table 1.

4.2. System framework and evaluation measures

The system framework, along with baseline system, was implemented in PyTorch, based on the available system implementations [1,2]. The input features were 23-dimensional log Mel-

[1] https://github.com/hitachi-speech/EEND

[2] https://github.com/Xflick/EEND_PyTorch

Table 2: *Diarization Error Rate (DER) results for the evaluated systems. The value of I indicates the number of refinement iterations applied consistently during system training and test phases. The test set for simulated Sim2Spk data includes four different levels of speech overlap in %. Values in the parentheses indicate the Miss (MI), False Alarm (FA) and Confusion (CF) errors.*

Model	Sim2Spk				CH
	47.3%	35.3%	28.0%	19.6%	
EEND-EDA	3.89 (2.1/1.7/0.2)	3.62 (2.2/1.1/0.3)	3.21 (1.7/1.1/0.4)	3.07 (1.8/1.0/0.3)	9.24 (5.5/2.7/1.0)
EEND-NAA, $I = 1$	4.42 (2.4/1.7/0.3)	4.09 (2.6/1.1/0.3)	3.52 (2.1/1.0/0.4)	3.31 (2.2/0.8/0.3)	8.94 (5.9/2.1/0.9)
EEND-NAA, $I = 2$	3.95 (2.1/1.6/0.2)	3.53 (2.4/1.0/0.2)	3.25 (2.0/1.1/0.2)	3.42 (1.9/1.2/0.3)	8.19 (4.8/2.5/0.8)
EEND-NAA, $I = 3$	3.59 (2.0/1.5/0.2)	3.18 (2.1/0.8/0.2)	2.97 (1.8/0.9/0.2)	**2.92** (1.4/1.2/0.3)	8.10 (4.5/2.8/0.7)
EEND-NAA, $I = 4$	3.37 (1.9/1.3/0.2)	3.15 (2.1/0.9/0.2)	2.90 (1.7/0.9/0.3)	3.46 (1.8/1.4/0.3)	7.94 (4.4/2.7/0.8)
EEND-NAA, $I = 4 + \mathcal{L}_{\mathrm{spk}}$	**3.23** (1.8/1.3/0.2)	**2.97** (1.9/0.9/0.2)	**2.77** (1.5/1.0/0.3)	3.39 (2.0/1.2/0.3)	**7.83** (4.3/2.7/0.8)

filterbank coefficients, computed with context of 7. Furthermore, sub-sampling by 10 and feature time-shuffling were applied.

The Transformer encoder was built of 4 layers, with 4-head attention mechanism, 2048 dimension in feed-forward layers, producing 256-dimensional embeddings. The decoder was built of 2 layers, with the same parameters as in the encoder.

During training, Adam optimizer was used with learning rate scheduler as in [20] with 100 000 warm-up steps. The networks where trained for 100 epochs. For each studied system, the final model was obtained by averaging models from 10 consecutive epochs, with the epoch number selected based on the validation loss values for that system. This procedure was adopted to avoid network over-fitting to the training data, which otherwise would be the case for the proposed model. During fine-tuning for the evaluation of the CALLHOME dataset, Adam optimizer was also used during training for 25 epochs, with the learning rate equal to 10^{-5}. The final model was averaged from models from 5 last epochs. Minibatch size of 64 was used. In experiments with speaker loss, we used margin equal to 0.3, scale 30 and set $\lambda = 0.001$.

Experiments were conducted in oracle scenario. As evaluation metric we used the standard Diarization Error Rate (DER) metric, with 0.25 s collar tolerance.

5. Results and discussion

Table 2 presents the DER results of performed experiments, along with the Missed Detection, False Alarm and Confusion errors. The first row represents the baseline EEND-EDA system, while the second row represents the EEND-NAA system without iterative refinement (i.e. with a single iteration $I = 1$). Comparing these two systems, we can observe that for the simulated scenario, the EEND-NAA does not provide an improvement over baseline, except for the CALLHOME set.

The block of the next three rows presents the results for the EEND-NAA system with an increasing number of refinement steps ($I = 2, 3$ and 4) applied during both training and test phases. As can be observed, the application of iterative refinement of system decisions leads to a clear gain in DER results. In particular, DER improves steadily with an increasing iteration number for the CALLHOME and three test sets with high speech overlap. In Figure 3, we also present T-SNE visualization of encoder embeddings labeled by system EEND-NAA with $I = 4$ at each iteration. As can be seen, during the first estimation, the system recognizes mainly overlap and silence. With each consecutive iteration, the system corrects its decisions, visible especially in iterations 1-3.

The last row presents the gain achieved when, in addition, the speaker classification loss is applied. As can be observed, for all test sets, the incorporation of additional speaker loss improves the DER results over the analogous system without such a regularization. Moreover, from the results presented in [11] for different number of speakers, we can expect further improvement in DER results for scenarios in which more speakers occur in the recording.

For the proposed system, a consistent improvement in DER was observed in almost all cases, except for the dataset with $\beta = 5$ and the lowest overlap (19.6%), for which the best performing system is EEND-NAA with $I = 3$. We hypothesise that degradation in DER is caused by relatively high Miss and False Alarm errors. Interestingly, for the CALLHOME set, which also exhibits low speech overlap, this discrepancy is not observed. Speech statistics for all datasets used during tests is presented in Table 1. As can be observed, the set with the lowest overlap is also characterized by the lowest number of speech frames (56%) in general, while CALLHOME has 90%. We suspect that a large proportion of silence frames impact negatively in our system and plan to work on a solution to handle this issue.

Finally, we observed that iterative refinement results in faster network convergence. For systems with refinement, the training procedure from the baseline system was redundant, as at that time the network already over-fitted to the training data.

6. Conclusions and outlook

In this paper, we presented a novel approach for system back-end in end-to-end speaker diarization, in which we replace the LSTM-based encoder-decoder attractor with a novel Transformer-based non-autoregressive approach. Furthermore, we incorporate an iterative refinement of system decisions and attractors; besides of adding speaker classification loss in the training objective. This method consistently improved over the baseline for various synthetic and CALLHOME datasets.

An interesting direction of future research will be to extend the method to scenarios with a higher yet unknown number of speakers and incorporate an additional mechanism to more robustly deal with non-speech segments.

7. Acknowledgements

This research received financial support from the National Science Centre of Poland under grant number DEC-2017/25/B/ST7/01792 and from the Foundation for Polish Science under grant number First TEAM/2017-3/23 (POIR.04.04.00-00-3FC4/17-00) which is co-financed by the European Union under the European Regional Development Fund, and was supported in part by PLGrid Infrastructure.

8. References

[1] S. H. Shum, N. Dehak, R. Dehak, and J. R. Glass, "Unsupervised Methods for Speaker Diarization: An Integrated and Iterative Approach," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 10, pp. 2015–2028, 2013.

[2] D. Garcia-Romero, D. Snyder, G. Sell, D. Povey, and A. McCree, "Speaker diarization using deep neural network embeddings," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 4930–4934.

[3] Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, and S. Watanabe, "End-to-End Neural Speaker Diarization with Permutation-Free Objectives," in *Proc. Interspeech 2019*, 2019, pp. 4300–4304.

[4] Y. Fujita, N. Kanda, S. Horiguchi, Y. Xue, K. Nagamatsu, and S. Watanabe, "End-to-End Neural Speaker Diarization with Self-attention," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 296–303.

[5] Y. Fujita, S. Watanabe, S. Horiguchi, Y. Xue, and K. Nagamatsu, "End-to-End Neural Diarization: Reformulating Speaker Diarization as Simple Multi-label Classification," *arXiv preprint arXiv:2003.02966*, 2020.

[6] S. Horiguchi, Y. Fujita, S. Watanabe, Y. Xue, and K. Nagamatsu, "End-to-End Speaker Diarization for an Unknown Number of Speakers with Encoder-Decoder Based Attractors," in *Proc. Interspeech 2020*, 2020, pp. 269–273.

[7] S. Horiguchi, S. Watanabe, P. García, Y. Xue, Y. Takashima, and Y. Kawaguchi, "Towards Neural Diarization for Unlimited Numbers of Speakers Using Global and Local Attractors," *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 98–105, 2021.

[8] T.-Y. Leung and L. Samarakoon, "Robust End-to-End Speaker Diarization with Conformer and Additive Margin Penalty," in *Proc. Interspeech 2021*, 2021, pp. 3575–3579.

[9] Y. C. Liu, E. Han, C. Lee, and A. Stolcke, "End-to-End Neural Diarization: From Transformer to Conformer," in *Proc. Interspeech 2021*, 2021, pp. 3081–3085.

[10] K. Kinoshita, M. Delcroix, and N. Tawara, "Integrating end-to-end neural and clustering-based diarization: Getting the best of both worlds," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7198–7202.

[11] K. Kinoshita, M. Delcroix, and T. Iwata, "Tight Integration Of Neural- And Clustering-Based Diarization Through Deep Unfolding Of Infinite Gaussian Mixture Model," *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8382–8386, 2022.

[12] K. Kinoshita, M. Delcroix, and N. Tawara, "Advances in Integration of End-to-End Neural and Clustering-Based Diarization for Real Conversational Speech," in *Proc. Interspeech 2021*, 2021, pp. 3565–3569.

[13] N. Chen, S. Watanabe, J. Villalba, P. Żelasko, and N. Dehak, "Non-Autoregressive Transformer for Speech Recognition," *IEEE Signal Processing Letters*, vol. 28, pp. 121–125, 2021.

[14] E. G. Ng, C.-C. Chiu, Y. Zhang, and W. Chan, "Pushing the Limits of Non-Autoregressive Speech Recognition," *Proc. Interspeech 2021*, pp. 3725–3729, 2021.

[15] Y. Higuchi, H. Inaguma, S. Watanabe, T. Ogawa, and T. Kobayashi, "Improved mask-CTC for non-autoregressive end-to-end ASR," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 8363–8367.

[16] W. Chan, C. Saharia, G. Hinton, M. Norouzi, and N. Jaitly, "Imputer: Sequence modelling via imputation and dynamic programming," in *International Conference on Machine Learning*. PMLR, 2020, pp. 1403–1413.

[17] J. Lee, E. Mansimov, and K. Cho, "Deterministic Non-Autoregressive Neural Sequence Modeling by Iterative Refinement," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 1173–1182.

[18] E. A. Chi, J. Salazar, and K. Kirchhoff, "Align-Refine: Non-Autoregressive Speech Recognition via Iterative Realignment," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 1920–1927.

[19] 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)*. IEEE, 2017, pp. 241–245.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is All You Need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Curran Associates Inc., 2017, p. 6000–6010.

[21] 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)*, 2019, pp. 4685–4694.

[22] D. Snyder, G. Chen, and D. Povey, "MUSAN: A Music, Speech, and Noise Corpus," 2015, arXiv:1510.08484v1.

[23] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5220–5224, 2017.

[24] M. Przybocki and A. Martin. (2001) 2000 NIST Speaker Recognition Evaluation LDC2001S97. Web Download. Philadelphia: Linguistic Data Consortium.

IV End-to-End Neural Speaker Diarization with Non-Autoregressive Attractors

M. Rybicka, J. Villalba, T. Thebaud, N. Dehak and K. Kowalczyk, "*End-to-End Neural Speaker Diarization with Non-Autoregressive Attractors*", IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2024.

End-to-End Neural Speaker Diarization With Non-Autoregressive Attractors

Magdalena Rybicka, Jesús Villalba, *Member, IEEE*, Thomas Thebaud, *Member, IEEE*, Najim Dehak, *Senior Member, IEEE*, and Konrad Kowalczyk, *Senior Member, IEEE*

***Abstract*—Despite many recent developments in speaker diarization, it remains a challenge and an active area of research to make diarization robust and effective in real-life scenarios. Well-established clustering-based methods are showing good performance and qualities. However, such systems are built of several independent, separately optimized modules, which may cause non-optimum performance. End-to-end neural speaker diarization (EEND) systems are considered the next stepping stone in pursuing high-performance diarization. Nevertheless, this approach also suffers limitations, such as dealing with long recordings and scenarios with a large (more than four) or unknown number of speakers in the recording. The appearance of EEND with encoder-decoder-based attractors (EEND-EDA) enabled us to deal with recordings that contain a flexible number of speakers thanks to an LSTM-based EDA module. A competitive alternative over the referenced EEND-EDA baseline is the EEND with non-autoregressive attractor (EEND-NAA) estimation, proposed recently by the authors of this article. NAA back-end incorporates k-means clustering as part of the attractor estimation and an attractor refinement module based on a Transformer decoder. However, in our previous work on EEND-NAA, we assumed a known number of speakers, and the experimental evaluation was limited to 2-speaker recordings only. In this article, we describe in detail our recent EEND-NAA approach and propose further improvements to the EEND-NAA architecture, introducing three novel variants of the NAA back-end, which can handle recordings containing speech of a variable and unknown number of speakers. Conducted experiments include simulated mixtures generated using the Switchboard and NIST SRE datasets and real-life recordings from the CALLHOME and DIHARD II datasets. In experimental evaluation, the proposed systems achieve up to 51% relative improvement for the simulated scenario and up to 15% for real recordings over the baseline EEND-EDA.**

***Index Terms*—Attractor mechanism, clustering, end-to-end, iterative refinement, non-autoregressive model, self-attention, speaker diarization.**

I. Introduction

SPEAKER diarization aims to answer the question "who spoke when" in a given utterance. It recognizes the segments where the same speaker occurs, identifying the silence and overlapping regions. Current diarization aims to solve various problems connected with real-life applications, from 2-speaker only conversations to cocktail party scenarios. However, that also comes with challenges that the systems have to deal with, e.g., long recordings, overlapping speech, or estimation of the number of speakers in the recording–which becomes more difficult as this number grows.

The most straightforward and well-established approach to deal with the diarization problems are cluster-based methods [1], [2], [3], which consist of a pipeline of several independent steps. Firstly, the recording is processed by a Voice Activity Detection (VAD) module in order to detect and remove from further analysis the silence regions. Next, the processed recording is divided into short overlapping segments, and from each such segment, a speaker embedding representation is extracted. Obtained embedding vectors are clustered, producing the final diarization decision. However, as each module is independent, the system may not be optimized properly for the overall diarization problem. Furthermore, such an approach is not feasible to handle properly the segments that contain overlapped speech and requires the application of additional mechanisms and post-processing. Several works attempt to deal with overlapped speech in the cluster-based methods [4], [5], [6]. In [4] the authors correct the system's diarization decision by processing the recording in two steps. First, a standard cluster-based diarization is performed, which returns the decision which speaker is most likely to be active in a particular frame. In the next step, the recording is processed to detect regions with speech overlap, so that in these regions the second most likely speaker is included in the diarization result too. In [6] the authors propose end-to-end overlap-aware re-segmentation, which further improves the results of the conventional diarization systems. A slightly different but effective approach to deal with overlap in the cluster-based diarization system is speech separation guided diarization (SSGD) [7],

Manuscript received 7 October 2023; revised 19 April 2024 and 28 June 2024; accepted 28 July 2024. Date of publication 7 August 2024; date of current version 9 September 2024. This work was supported in part by Fulbright Junior Research Award granted by Polish-U.S. Fulbright Commission, in part by the Foundation for Polish Science under Grant TEAM/2017-3/23, in part by the European Union through European Regional Development Fund, in part by National Science Centre, Poland under Grant 2021/42/E/ST7/00452, and in part by program "Excellence initiative – research university" for the AGH University of Krakow. The associate editor coordinating the review of this article and approving it for publication was Dr. Keisuke Kinoshita. *(Corresponding author: Magdalena Rybicka.)*

Magdalena Rybicka was with the Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD 21218 USA. She is now with the AGH University of Krakow, 30-059 Krakow, Poland (e-mail: mrybicka@agh.edu.pl).

Jesús Villalba, Thomas Thebaud, and Najim Dehak are with the Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD 21218 USA, and also with the Human Language Technology Center of Excellence, Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: jvillal7@jhu.edu; tthebau1@jhu.edu; ndehak3@jhu.edu).

Konrad Kowalczyk is with the AGH University of Krakow, 30-059 Krakow, Poland (e-mail: konrad.kowalczyk@agh.edu.pl).

Digital Object Identifier 10.1109/TASLP.2024.3439993

where the authors use diarization and speech separation in a single system. Speech separation combined with VAD module is used to enhance system capabilities in the overlap regions. This system achieved the first place in the DIHARD III challenge.

The End-to-End Neural Speaker Diarization (EEND) framework, first proposed in [8], addresses many of the aforementioned problems. As an end-to-end system, it is trained to directly optimize the diarization result. Furthermore, reformulating the training objective into the form of multiple binary classification tasks, i.e., the presence or absence of each speaker, allows us to easily identify the overlapping regions and produce separate speech/non-speech tracks for each speaker. An important development is the EEND with encoder-decoder attractor (EEND-EDA) [9] that enables diarizing a flexible number of speakers through additional estimation of the so-called attractors, i.e., utterance-level speaker representations.

EEND has become a strong baseline for diarization and a starting point for the proposal of other EEND-based diarization systems [10], [11], [12], [13]. Many recent advancements in diarization research focus on the combination of the advantages of EEND and clustering-based systems by incorporating unsupervised clustering methods in the end-to-end pipeline, such as in the EEND with vector clustering (EEND-VC) system proposed in [10], [14], [15], with the aim of complementing the limitations of each of these approaches. EEND-VC operates on recording chunks, instead of the whole recording, and outputs not only the diarization results but also speaker-level representation corresponding to each track. In order to combine the tracks of the same speaker but from different chunks, the clustering algorithm is employed.

In addition to the EEND approach, another important study in the diarization area is the Target-Speaker Voice Activity Detection (TS-VAD) [16]. The idea behind the TS-VAD-based system is straightforward - using pre-computed (enrollment) speaker i-vector representations [17], TS-VAD extracts the voice activities for each speaker. In recent developments [11], [12], the ideas from the TS-VAD system are combined with EEND. In [11] the authors present the so-called EDA-TS-VAD, which attempt to combine the TS-VAD system with EEND-EDA. They argue that the EEND-EDA dot-product operation may not be suitable for time frames with overlapped speech. This system follows the EEND-EDA frame-level and attractor generation, replacing the EEND-EDA dot product operation with the TS-VAD-inspired Joint-Speaker-Detection block (JSD). The JSD block concatenates frame-level and utterance-level representations and processes cross-time and cross-speaker relations to obtain the diarization decisions. Another work that combines EEND-EDA with TS-VAD is Attention-based Encoder-Decoder network for End-to-End Neural Speaker Diarization (AED-EEND) [12]. The authors incorporate the EEND-EDA framework and replace the LSTM-based module with the attention-based decoder. Similarly to TS-VAD, they use enrollment information to obtain speaker representations. Contrary to the TS-VAD, this information is obtained by the model itself, not by the external system.

In this article, we propose End-to-End Neural Speaker Diarization with Non-Autoregressive Attractors (EEND-NAA). Our work can be related to developments presented in the literature, but also proposes a different perspective and derives its inspiration from Automatic Speech Recognition (ASR) [18], [19], [20], [21], where non-autoregressive methods are a competitive alternative to the autoregressive ones. Our proposed system uses a clustering approach but follows the EEND-EDA framework and end-to-end pipeline, where the autoregressive LSTM-based backend is replaced with non-autoregressive attractor estimation. More importantly, our proposal allows to make the process of attractor generation explainable, while the LSTM-based is more obscure. The idea of EEND-NAA has initially been proposed by the current authors in [13], where the EDA module is replaced with a simple initial attractor estimation using a k-means clustering algorithm applied on top of the frame-level embeddings, refined with attention-based decoder layers and an iterative refinement from the diarization system output itself. Our work [13] was the first to propose the use of non-autoregressive attractor estimation for the diarization task. However, the evaluation of the EEND-NAA in [13] is restricted to only a known and limited number of speakers, specifically to recordings with only 2 speakers. In this article, we aim to extend EEND-NAA system capabilities to generic conditions where the speaker number can vary between the recordings and may even be unknown. We extend the NAA module to include two consecutive decoder blocks and introduce an additional Single Speaker Activity Detection (SSAD) module that allows to identify frames which include only one active speaker and replaces the iterative refinement mechanism introduced in [13]. The proposed modifications lead to the design of three novel EEND-NAA variants capable of performing diarization of an unknown number of speakers in an end-to-end fashion.

In Section II, we first present a short overview of the related works in the diarization research. Next, in Section III the baseline EEND with EDA back-end is described. Sections IV and V introduce the original EEND-NAA system along with three possible extensions proposed in this article. Sections VI, VII, and VIII present the experimental evaluation setup, obtained results, and conclusions.

II. Related Work

A. End-to-End Neural Speaker Diarization

End-to-End Neural Diarization (EEND) [8] has originally been proposed as a BLSTM-based encoder that transforms the input sequence of features to the frame-level embedding representations, followed by a simple binary classification layer with the number of outputs equivalent to the maximum possible number of speakers in the recording. Each output is responsible for generating a separate potential speaker track. The training objective is defined as a multi-label classification problem, indicating whether a speaker occurs in a particular frame and track. Since the order of speaker tracks returned by the network is not predetermined, it raises the question of how to assign the ground truth labels. The problem is solved using the permutation-invariant training (PIT) loss [22]. The diarization output is compared with each label order permutation, and the assignment that results in the lowest loss value is selected as the correct one. In its development, the BLSTM encoder is replaced with the self-attention-based encoder [23], [24]. The next important generation of EEND systems is EEND with encoder-decoder attractor (EEND-EDA), which allows to deal with various and unknown number of speakers. The proposed EDA module estimates the number of speakers that occur in the recording and produces an utterance-level embedding representation for each speaker. In [25], the authors develop the EEND-EDA system to tackle the problem of large speaker numbers unseen during training with global and local attractors (the so-called EEND-GLA model), and adding clustering to the framework. Firstly, the embedding sequence is segmented into shorter chunks. Each subsequence is processed by the EDA back-end, producing corresponding attractor and diarization results. In order to obtain the result for the entire recording, the clustering is applied on top of the attractors. In [26] the authors present an extended evaluation and further improvement of the original EEND-EDA system by a modification of the training procedure and a proposal of a mechanism to deal with the number of speakers that is unseen during the training.

B. End-to-End Neural Diarization With Vector Clustering

An important EEND alteration is the EEND vector clustering (EEND-VC) system proposed in [10], [14], [15], which represents a hybrid approach that aims to combine the advantages of classical EEND [23] and cluster-based diarization. The original EEND system requires a large amount of memory to process long recordings. EEND-VC leverages the problem by segmenting the recording into smaller chunks and applying EEND-based diarization on each one of them. Within each chunk, each output track has an associated speaker representation. The system has an important assumption that the processed segment, typically of up to 50 s duration, has up to 2-3 speakers so that the simple EEND with classification output can be applied. All speaker representations are clustered together, which indicates which tracks from different chunks originate from the same speaker. A similar approach is proposed in [27], which, however, is based on the EEND-EDA [9] instead of the classical EEND [23]. The processed recording is segmented into smaller chunks, resulting in a diarization decision and attractor estimation from the EEND-EDA system per each chunk. The final diarization decision for the entire recording is obtained by combining the speaker tracks from different chunks. The decision of which track belongs to the same speaker is conducted by clustering the attractor representations using neural clustering based on the Gated Recurrent Units (GRU).

An important continuation of the work on the EEND-VC system is the Graph-PIT-EEND-VC [28], which performs the so-called utterance-by-utterance diarization. The originally proposed EEND-VC system shows promising results. However, it struggles with the strict assumption that the analyzed segment can contain a fixed, small amount of speakers, and at the same time, it is constrained from increasing the possible time context. Moreover, the authors argue that splitting audio into fixed-size chunks is impractical for the downstream tasks and can also result in very short speech segments. As a solution, the authors of [28] propose to retrieve speech segments with a 2-channel output VAD, where overlapping speakers are separated into different channels. The embeddings are computed using information from the continuous (non-chunked) speech segments of a particular speaker. Then, similarly to the original EEND-VC, speaker representations are clustered to decide which segments belong to the same speaker. The idea allows to increase the analyzed time context to the whole utterance and avoid its segmentation into short chunks, which was needed to meet the assumption of a small speaker number in the processed fragment.

The approach may seem to be similar to ours as it also incorporates the clustering approach on top of the EEND encoder. The main goal of using clustering in the EEND-VC-based systems is to find which segments or chunks belong to the same speaker based on the speaker representations of those chunks/segments. Note that it is different from our approach, where clustering is used to estimate the initial attractor representations from the entire utterance based on the frame-level embeddings.

C. Other Non-Autoregressive Approaches

After our proposal of EEND-NAA in [13], other works have appeared that also incorporate non-autoregressive processing in a diarization system. In [29] the authors propose adding an intermediate attractor representations in between EEND transformer encoder layers in order to condition subsequent layers. Similarly to our work, they replace LSTM-based EDA module with Transformer Decoder layer. However, as the initial attractor queries, learnable vectors are incorporated in [29]. The system limitation is that it does not handle speaker counting and results are presented only for 2-speaker recordings. In AED-EEND [12], already mentioned in Section I, similar to our work, the authors incorporate Transformer Decoder based EDA module. During inference, the system decodes the speakers iteratively: firstly, using trainable vector queries, the system detects the non-speech, overlap, and single-speech regions. Next, using the single speaker regions, it extracts speaker representations one by one, discovering the speaker existence of the new speakers. This is the main difference from our approach where we use clustering directly to estimate attractors, which we do in a single iteration pass.

III. Baseline End-to-End Neural Diarization With Encoder-Decoder Attractor

In this section, we present the general framework of the EEND system and describe the EEND-EDA processing [9], which constitutes the baseline for our approach and the presented evaluations. The main building blocks of the EEND system are the *encoder*, which produces the frame-level embedding representations, and the *backend*, which processes the embedding sequence and estimates the utterance-level speaker representations, i.e., attractors. The general scheme of the EEND-EDA is presented in Fig. 1.

The EEND input is represented by a sequence of features $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$ of length T and feature dimension F. The encoder generates a sequence of D-dimensional embeddings

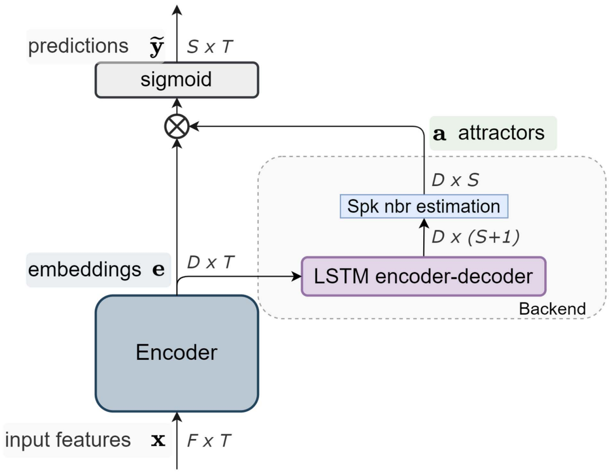


Fig. 1. General pipeline scheme of the EEND-EDA system. F indicates feature dimension, D is an embedding dimension, S stands for speaker number in the recording, and T is the number of time frames.

$\mathbf{e} = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_T\}$ of the same length, where each embedding corresponds to a single input feature. The encoder is built of four stacked Transformer encoder layers with a self-attention mechanism and does not include positional encoding.

The embedding representations are passed to the EDA module, which serves as the system back-end. The EDA generates attractors, the number of which is assumed to be equivalent to the total number of speakers in a given utterance. This block consists of two LSTM layers that are connected in an encoder-decoder manner. Theoretically, the EDA module can generate an infinite number of representations. In order to distinguish and constrain whether a produced attractor represents a subsequent speaker track, a linear layer with a sigmoid activation is situated at the top of the EDA module. The result returned by the layer is used for a decision to determine whether a particular embedding attractor represents a new speaker or the attractor generation process is finished. Based on the output of the binary classification layer, the attractor loss $\mathcal{L}_{\text{ext}}$ is computed. It is represented by binary cross-entropy loss. The layer output is compared to the labels, which are $S+1$ binary values, where S represents the speaker number in a given utterance. The first S labels are set to 1, while $(S+1)$-th label is set to 0. Such a label arrangement specifies the stopping condition of the attractor generation.

In the final step, the obtained attractors and each of the frame-level embeddings are used to compute a dot product. The resulting score, passed through the sigmoid function σ, represents the final diarization result, which can be equivalently presented as

$$\widetilde{y}_{s,t} = \sigma(\mathbf{e}_t \cdot \mathbf{a}_s) \ , \tag{1}$$

where $\widetilde{y}_{s,t}$ represents the posterior probability for speaker s to be present in time frame t, $\mathbf{e}_t$ is the frame-level embedding for frame t, and $\mathbf{a}_s$ is the attractor for the s-th speaker track. The score is also used to compute system's diarization loss $\mathcal{L}_d$, which is based on the binary cross-entropy loss. The final output of the system is compared with binary labels $\mathbf{y} \in \{0,1\}^{S \times T}$, where $y_{s,t} = 1$ indicates that speaker s is present at time frame t and $y_{s,t} = 0$ indicates that it is not present in the t-th time frame. The permutation invariant training (PIT) scheme [22] is used to determine the optimal mapping between the predicted and ground truth labels and compute the final diarization loss. The reader is referred to [9] for a more detailed description of the loss computation. The final training objective is composed as a sum of the attractor and diarization losses:

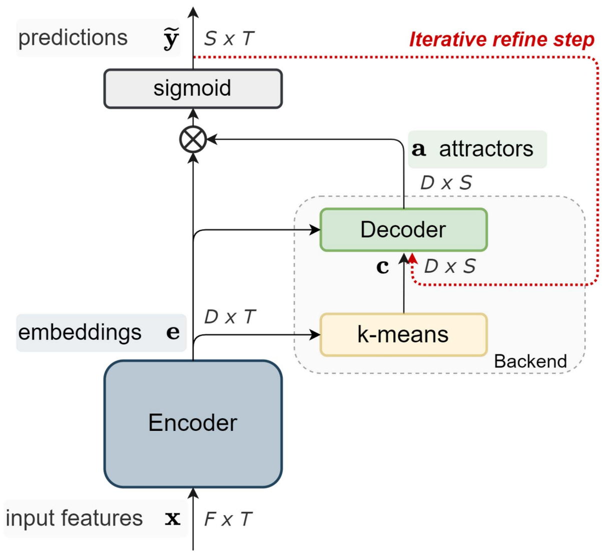


Fig. 2. General pipeline scheme of the EEND-NAA-Fixed. The k-means initialization is used during the first iteration. In the next iterations, the diarization result from the previous iterative refinement step is used. F indicates feature dimension, D is an embedding dimension.

$$\mathcal{L} = \mathcal{L}_d + \alpha \mathcal{L}_{\text{ext}}, \tag{2}$$

where α is a weight applied to the attractor loss.

IV. EEND-NAA for a Known Number of Speakers

This section presents a detailed description of the EEND system with non-autoregressive attractor estimation, introduced for the first time in our conference paper [13]. Note that the results presented in [13] include only the oracle condition for 2-speaker recordings. As the presented system can handle only a known number of speakers, we will refer to this model version as EEND-NAA-Fixed. Later, in Section V, we will develop and introduce further EEND-NAA system modifications that enable the application of the system for non-oracle and a variable speaker number in the recording.

A. Non-Autoregressive Attractor Estimation

Block diagram of the EEND-NAA-Fixed system is presented in Fig. 2. The processing flow is similar to the EEND-EDA system, where the encoder, composed of the Transformer encoder layers, transforms input features into frame-level embedding representations. The obtained embedding sequence is processed by the back-end to extract utterance-level attractor embeddings. The initial attractor embeddings are estimated using the k-means algorithm applied on top of the frame-level embeddings. The initial vectors are the cluster centers, denoted hereafter as $\mathbf{c}_s \in$

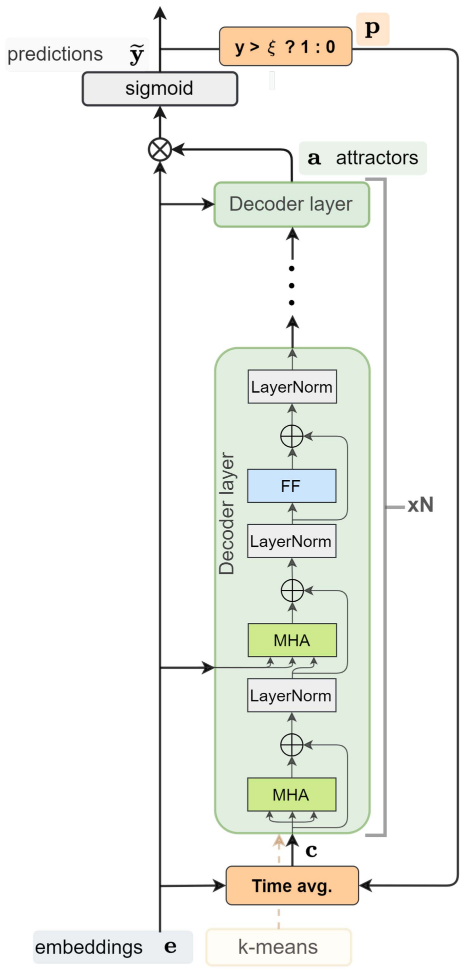


Fig. 3. Detailed layer architecture of one of N of the EEND-NAA transformer-based backend blocks.

$\{\mathbf{c}_1, .., \mathbf{c}_S\}$. The number of clusters corresponds to the number of speakers in the recording.

Obtained initial vectors are next refined by a decoder block composed of 2 layers of Transformer decoder [30]. Fig. 3 illustrates the diagram structure of the N-layer decoder back-end block. A Transformer decoder layer consists of three essential layers: a multi-head self-attention layer, a multi-head source-target attention layer, and a position-wise feedforward layer (FF). Both attention layers are referred to as MHA blocks. Each layer is followed by a residual connection, which combines the input and output of the layer, and a normalization layer. The core components of the attention mechanism are the key, value, and query linear transformations. The processing flow of both self-attention and source-target attention is identical with the difference in the source of the input to these elements. Self-attention processes only the estimated cluster centers or input from the previous decoder block, whereas source-target attention involves embedding representations to compute the key and value, and intermediate attractor estimations act as queries. The positional encoding is not used.

Following the EEND-EDA, the final attractors and embeddings are used to compute the dot product, and after processing with the sigmoid function, the final diarization result is produced. The loss used in the EEND-NAA-Fixed system training incorporates only the diarization loss with the PIT method.

B. *Iterative Refinement*

As explained in [13], the goal of using k-means at the top of the embedding sequence is to estimate general initial representations of the attractors which should be refined by the decoder. However, such an approach suffers from two drawbacks. The clustering is applied on top of all embeddings, not only speaker representations but also silence embeddings. Moreover, it is applied to embeddings containing overlapped speech, where such embeddings should belong to more than one cluster (corresponding to more than one speaker). To deal with the aforementioned issues, we apply the iterative refinement of the estimated attractors. It means that the representations are recalculated using the diarization system output and reprocessed by the decoder, which allows us to discard silence embeddings and include overlap in the computation of the new means.

Once the attractors are initially estimated using the k-means algorithm, diarization results $\widetilde{y}_{s,t}$ are iteratively obtained by computing the dot product between the attractors and the encoder embeddings, with the application of the sigmoid function σ, as given by

$$\widetilde{y}^{i}_{s,t} = \sigma(\mathbf{e}_t \cdot \mathbf{a}^{i}_{s})\ , \tag{3}$$

where i denotes the iteration number, and $\mathbf{a}$ is the attractor representation. To obtain the binary diarization decision, we compare the resulting value for each diarization track with the neutral threshold value of $\xi = 0.5$ using

$$p^{i}_{s,t} = \begin{cases} 1 & \widetilde{y}^{i}_{s,t} > \xi \\ 0 & \widetilde{y}^{i}_{s,t} \leq \xi \end{cases} . \tag{4}$$

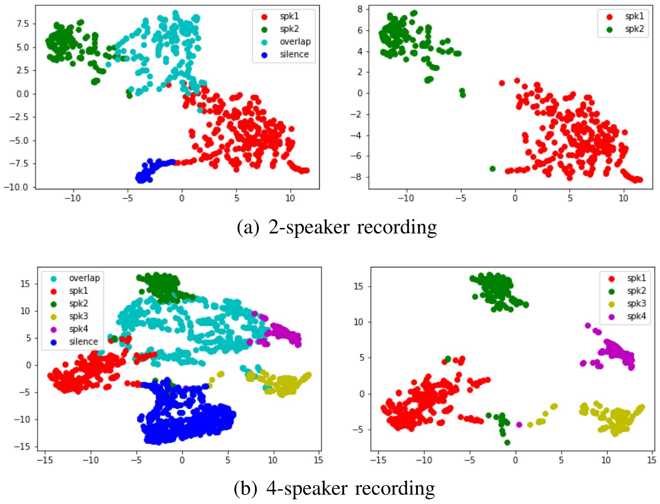
Based on the obtained diarization result, new initial cluster centers are computed according to

$$\mathbf{c}^{i}_{s} = \frac{1}{\sum_{t=1}^{t=T} p^{i-1}_{s,t}} \sum_{t=1}^{t=T} p^{i-1}_{s,t} \cdot \mathbf{e}_t\ . \tag{5}$$

The centers are again passed through the decoder stack to estimate the refined attractors $\mathbf{a}^{i}_{s}$. Note that this center recalculation uses the diarization decision, allowing us to incorporate, in the center computation, the overlap embeddings $\mathbf{e}_t$ for all speakers present at time frame t and discard the silence embeddings. It should also be noted that this process applies to all iterations with $i > 1$, whereas for the first iteration (for $i = 1$), the initial representations $\mathbf{c}$ are calculated using k-means clustering.

V. EEND-NAA for an Unknown Number of Speakers

While the EEND-NAA-Fixed system described in Section IV can achieve promising results [13], it is not feasible to use it in scenarios where the number of speakers is generally unknown. This limitation is a consequence of using k-means clustering, which requires the number of clusters (i.e., speakers) to be known in advance. To address this problem, in this section, we introduce three novel versions of the EEND-NAA system, which



(a) 2-speaker recording

(b) 4-speaker recording

Fig. 4. T-SNE visualization of the encoder embeddings obtained by EEND-NAA (with $I = 4$). Plots on the left include all frame-level embeddings, plots on the right contain only embeddings that contain a single speaker.

can handle the non-oracle number of speakers scenario in various ways.

The general model pipeline is kept intact compared to EEND-NAA-Fixed system, where the proposed adjustments are applied only in the system's back-end. All systems include a new Single Speaker Activity Detection (SSAD) module. The goal of this block is to filter the encoder embeddings that represent overlapped speech or silence regions and thereby keep for further processing only embeddings that contain speech from a single speaker. Note that, in a typical conversational scenario, we expect that the speakers of interest will have regions of uninterrupted, non-overlapped speech, while speakers that only speak while others are speaking (i.e., speakers that only appear in overlapped regions) will be rare and will not have an important contribution to the overall conversation. This assumption allows us to use the SSAD module to improve the cluster centers of those speakers of interest. The module comprises one Transformer encoder layer and a two-output linear classification layer. The problem is formulated as a multi-label classification. The applied SSAD module has two outputs, which correspond to silence/non-silence and overlap/non-overlap decisions. The other multi-class alternative could be a three-output SSAD in which each output corresponds to the single-speaker, silence and overlap classes. We have run preliminary experiments with two and three outputs and did not observe any notable difference in performance between such two models. Thus, as a design choice, the two-output model was chosen as a simpler variant. The SSAD block is crucial for the proposed system to facilitate detecting the number of speakers present in the recording. Note that the embeddings generated by the encoder generally tend to create clusters that represent the speakers and silence, while overlap embeddings are located in between those clusters. In Fig. 4 we present T-SNE visualization of the embeddings from the two different recordings from the test set used in experiments to follow. One recording has 2 speakers, the second contains 4 speakers. In both cases, we present T-SNE plots with all embeddings and with embeddings that contain only a single speaker in the particular frame. For both examples, we can observe similar embedding composition: the embeddings that belong to the same class, i.e. to any particular speaker, overlap, or silence, tend to be close to each other. Let us note that the overlap class is in fact a composition of multiple "single speaker classes", thus we can observe that it tends to be located in between of clusters for the respective single speakers. To obtain clear cluster representations, we must filter out the overlap embeddings that "contaminate" the clusters' centers. The methods presented in the following subsections strongly rely on the proper clustering of the frame-level embeddings and we assume that the cluster centers represent the preliminary speaker embeddings. Moreover, in the case of two out of three systems proposed in this section, the clustering algorithm is used to estimate the number of speakers. For this reason, any contamination by overlap embeddings or presence of additional (non-speaker) silence cluster affects the result of the clustering algorithm, and thus the overall system diarization decision. The additional consequence of introducing the SSAD module is that there is no need for incorporating the iterative refinement mechanism, as we do not have to deal with embeddings for silence and overlap.

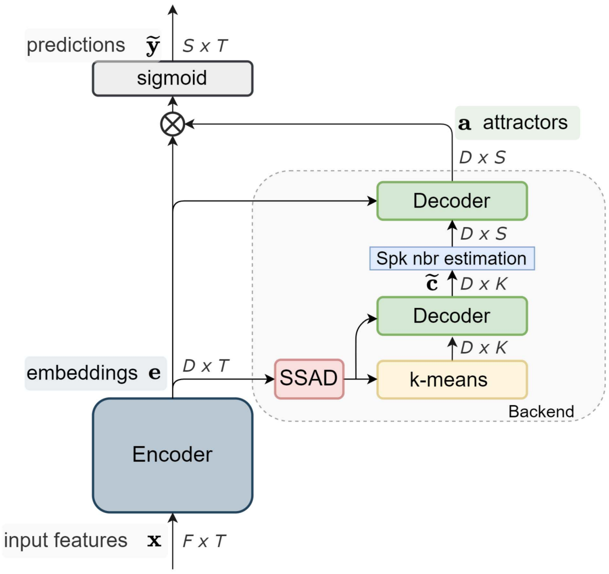


Fig. 5. General pipeline scheme of the EEND-NAA-Overest system. In the diagram, K stands for the chosen number of clusters, while S is the number of speakers in the recording, F indicates feature dimension, and D denotes embedding dimension.

A. EEND-NAA-Overest

The first presented system, hereafter referred to as EEND-NAA-Overest, is an initial attempt to extend the EEND-NAA system for the condition of an unknown number of speakers. The general system pipeline is shown in Fig. 5.

The primary modification to the EEND-NAA-Fixed system is the choice of the number of clusters for the k-means algorithm. As the number of speakers in the recording may be unknown, we intentionally overestimate the number of clusters. It means that the number of clusters used in the k-means algorithm is higher than the expected maximum number of speakers in the recording. The next system modification, compared with the EEND-NAA-Fixed, is the back-end extension from one to two decoders, where each decoder performs its own designated task. Both decoder blocks share the same structure, and they are composed of two transformer decoder layers.

The task of the first decoder is to process all estimated cluster centers and retrieve refined centers, correcting errors made by the k-means. In this block, only filtered embeddings, i.e., embeddings with speech from one speaker, are used as key and value inputs in the Transformer decoder layers.

To enforce the first decoder to produce speaker/non-speaker discriminative centers, we use the softmax cross-entropy loss, which we will refer to as a pre-attractor distance loss,

$$\mathcal{L}_a = -\frac{1}{S}\sum_{s=1}^{s=S}\log\frac{\exp(\cos(\mathbf{h}_s, \widetilde{\mathbf{c}}_s))}{\sum_{j=1}^{j=K}\exp\left(\cos(\mathbf{h}_s, \widetilde{\mathbf{c}}_j)\right)} \tag{6}$$

where the argument of the softmax function is the cosine score between the refined centers $\widetilde{\mathbf{c}}_\mathbf{s}$ and the "ideal" centers $\mathbf{h}_s$, K denotes the total number of clusters, while S denotes the number of speakers in the recording. The ideal centers are computed based on the ground truth labels as

$$\mathbf{h}_s = \frac{\sum_{t=1}^{t=T} z_{s,t}\cdot\mathbf{e}_t}{\sum_{t=1}^{t=T} z_{s,t}}\ , \tag{7}$$

where s is speaker index, $z_{s,t}$ is the ground truth label at time frame t for speaker s. Note that overlap embeddings are not included in the computations of the aforementioned mean. It means that $z_{s,t}$ value is equal to 1 if speaker s is present in a particular time frame t, and equal to 0 if it is not, or if there is more than one speaker in the time frame.

The assignment of cluster centers to particular speakers is realized through a permutation scheme. For each possible permutation of K cluster centers with size of S (note that S is smaller than K), we compute the sum of the distances between ideal centers and cluster centers before decoder processing. The option (i.e. permutation of a subset of K clusters with size of S) that results in the lowest value, is the cluster center assignment. Note, that the loss value targets to refine the relative, discriminative speaker representations within the recording, not the absolute speaker identity, as some other approaches do [10]. A similar goal of refinement is presented in [25], however, the loss used in [25] is based on the contrastive loss and distances between the local attractors.

The refined centers from this step serve as preliminary attractors (pre-attractors). A linear binary classification layer processes each pre-attractor to detect which ones correspond to true speakers, and which ones correspond to the overestimated clusters, which we should discard. The ground truth label defining whether pre-attractors are speakers or not, is derived from the best permutation obtained by the loss in (6). The selected pre-attractors are forwarded to the second decoder, which produces the final attractors, using all embeddings from the sequence as key and value inputs.

The final training objective for the EEND-NAA-Overest system can be presented as

$$\mathcal{L} = \mathcal{L}_d + \mathcal{L}_{\text{SSAD}} + \mathcal{L}_{\text{choice}} + \lambda\mathcal{L}_a\ , \tag{8}$$

where $\mathcal{L}_d$ is the diarization loss, calculated as binary cross-entropy loss with PIT scheme, in an identical way as described in Section III for the EEND-EDA system; $\mathcal{L}_{\text{SSAD}}$ is the SSAD classification cross-entropy loss, deciding whether the particular frame contains overlapped speech or silence; $\mathcal{L}_{\text{choice}}$ is the binary classification loss for the linear layer that decides whether the particular pre-attractor is the speaker or non-speaker representation; $\mathcal{L}_a$ denotes the pre-attractor distance loss, defined in (6); whilst λ is an empirically selected parameter.

B. EEND-NAA-2step

In the proposed EEND-NAA-2step approach, similarly to the previous EEND-NAA-Overest approach described in Section V-A, on top of the SSAD filtered encoder embeddings, we apply k-means clustering with a higher number of cluster centers than the maximum possible number of speakers in the recording. Let us note that it also means that embeddings that belong to one speaker can be included in a few cluster centers. Thus, one speaker can be represented by multiple cluster centers. Next, the centers are processed by the first decoder, whose task is slightly different than in the previous EEND-NAA-Overest system. The goal is to bring the cluster centers from the same speaker as close to each other as possible and, simultaneously, to push apart the centers from different speakers as far as possible. After that, we aim to merge the refined centers in order to obtain a single representation for each speaker. The centers are combined with a second clustering applied to the refined centers. We use k-means for the second clustering during training, while spectral clustering is used for inference. Spectral clustering produces better results but is computationally too expensive to be used in training. The number of speakers is estimated by the analysis of the eigenvalues [31]. Fig. 6 presents the general EEND-NAA-2step system scheme pipeline.

In order to teach the first decoder the designated task, the softmax contrastive loss is applied on top of the processed centers. For the EEND-NAA-2step system, the assignment of K centers to the specific speakers is based on the first clustering result, i.e., clustering before the first decoder. Based on the assignments of the frame-level embeddings to the particular cluster, we decide to which speaker the center belongs to. If the center is composed of the embeddings that belong to different speakers, the center is assigned to the speaker whose embeddings are mostly used to compute it.

Similarly, as in the previous system, the softmax loss is used. However, it is applied to each center separately in the following manner:

$$\mathcal{L}_a = -\frac{1}{K}\sum_{k=1}^{k=K}\log\frac{\exp(\cos(\mathbf{h}_k, \widetilde{\mathbf{c}}_k))}{\sum_{j=1, s_j\neq s_k}^{j=K}\exp\left(\cos(\mathbf{h}_k, \widetilde{\mathbf{c}}_j)\right)}. \tag{9}$$

Note that this is a contrastive loss rather than a cross-entropy loss since the sum of the denominator of (9) only has negative pairs–i.e., the centers that belong to the same speaker class as

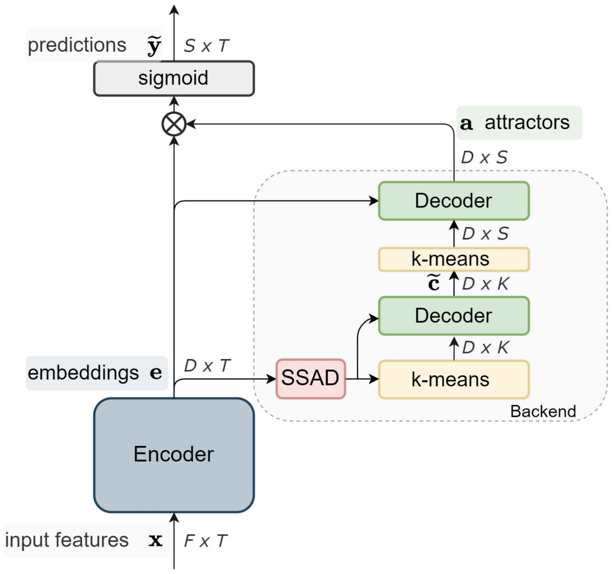


Fig. 6. General pipeline scheme of the EEND-NAA-2step system. In the diagram K stands for the chosen number of clusters, while S is the number of speakers in the recording. F indicates feature dimension, D embedding dimension.

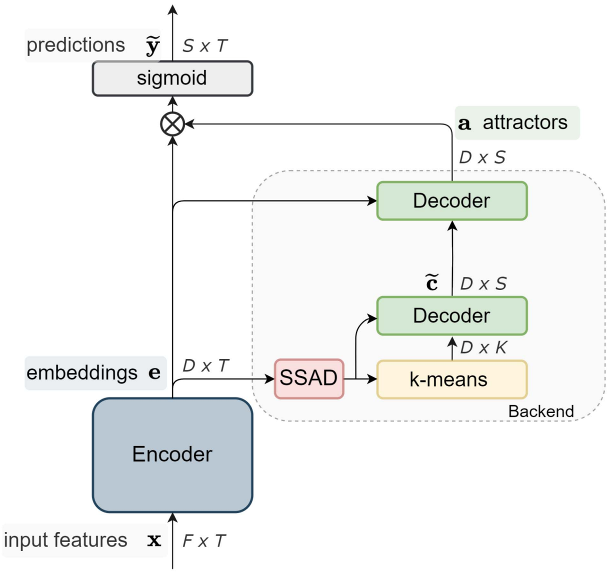


Fig. 7. General pipeline scheme of the EEND-NAA-1step system. In the diagram, K stands for the chosen number of clusters, while S is the number of speakers in the recording, F indicates feature dimension, and D denotes embedding dimension.

$\tilde{\mathbf{c}}_k$ are not used. Furthermore, this loss can only be applied if $S > 1$, because for $S = 1$, there are no negative pairs. In case $S = 1$, the softmax loss is replaced with the simple sum of cosine distances between each unmerged pre-attractor to the "ideal" centers. Then, the second k-means with $K = S$, merges the pre-attractors belonging to the same speaker. The merged representations are forwarded to the second decoder, which produces the final attractors. As the system applies clustering twice, we refer to it as the EEND-NAA-2step.

The final training objective, similar to the EEND-NAA-Overest system, can be formulated as

$$\mathcal{L} = \mathcal{L}_d + \mathcal{L}_{\text{SSAD}} + \lambda \mathcal{L}_a \ , \tag{10}$$

where $\mathcal{L}_d$ is the diarization loss, $\mathcal{L}_{\text{SSAD}}$ denotes the SSAD classification loss, $\mathcal{L}_a$ is a pre-attractor distance loss, while λ is an empirically selected parameter. Note that the loss of this system does not include the $\mathcal{L}_{\text{choice}}$ term.

C. EEND-NAA-1step

The last proposed system simplifies EEND-NAA-2step and, at the same time, is the most similar to the original EEND-NAA-Fixed system. Fig. 7 presents its pipeline. As in EEND-NAA-Fixed, only one clustering is used, so we refer to this system as EEND-NAA-1step.

Firstly, during the training phase, the k-means algorithm is applied at the top of the SSAD-filtered frame-level embeddings with $K = S$. During the inference, the k-means is replaced with spectral clustering, with eigenvalue analysis to estimate the number of speakers. The cluster centers are forwarded to the first decoder block. The goal of that block is to refine the centers to ideal representations by applying the softmax cross-entropy loss for recordings with $S > 1$ as in (6), and the cosine distance between the mean of the refined centers and the ideal representation for $S = 1$. Next, the refined centers are forwarded to the second decoder block, producing final attractor representations. The final objective function of the system is identical to that in (10).

TABLE I
NUMBER OF PARAMETERS OF THE EEND-EDA AND EEND-NAA STRUCTURES

System	#Parameters
EEND-EDA	6.4 M
EEND-NAA-Fixed	8.5 M
EEND-NAA-Overest	13.0 M
EEND-NAA-1step	13.0 M
EEND-NAA-2step	13.0 M

In Table I we present the number of parameters for presented systems which shows the impact of the additional Transformer layers on the model size. The EEND-NAA-Fixed has 2.1M more parameters, where the newly proposed systems have nearly 2 times more parameters than EEND-EDA reference. Nevertheless, we still consider this model size as a reasonable.

VI. Experimental Evaluation

A. Datasets

The training set is composed of simulated mixtures, generated in the same manner as proposed in [24], with the use of utterances from the datasets: Switchboard-2 (Phase I, II, III), Switchboard Cellular (Part 1 and 2), and the NIST Speaker Recognition Evaluation (2004, 2005, 2006, 2008). The produced mixtures contain from 1 to 4 speakers within each recording, with 100 000 utterances per each speaker number. The overlap for the 2 / 3 / 4 speaker recordings is, respectively, equal to 34.4% / 34.8% / 32.0%. The experiment presented in Section VII-A uses only 2 speaker recordings in the training phase (which results in

TABLE II
STATISTICS OF THE SIMULATED DATASETS

Subset	#Spk	#Mixtures	#Hours	Overlap ratio, %
Train				
	1	100 000	2 155	0.0
	2	100 000	2 480	34.7
	3	100 000	4 221	34.8
	4	100 000	6 645	32.0
Test				
	1	500	11	0.0
	2	500 / 500 / 500	12 / 15 / 20	35.3 / 28.0 / 19.6
	3	500	21	35.3
	4	500	33	32.7

TABLE III
STATISTICS OF THE DATASETS CONTAINING REAL RECORDINGS

Dataset	#Spk	#Mixtures	#Hours	Overlap ratio, %
CALLHOME				
Part 1	2-7	249	9	17.0
Part 2	2-6	250	9	16.7
DIHARD II				
dev	1-10	192	24	9.8
eval	1-9	194	22	8.9

100 000 utterances), while the results presented in Section VI-I-B incorporate all mixtures for each speaker number (which results in 400 000 utterances). Table II presents the details of the generated simulated dataset.

System evaluation includes both simulated mixtures and real-life recordings. For the experiment with a fixed number of 2 speakers, 3 sets of 500 utterances with different speech overlaps (35.3%, 28.0%, and 19.6%) are used. The test set for the multi-speaker experiment (described in Section VII-B) is composed of 4 sets of 500 utterances each, with the number of speakers varying from 1 to 4. The speech overlap for 2 / 3 / 4 speakers, respectively, amounts to 35.3% / 35.3% / 32.7%. All train and test simulated mixtures are augmented with noise from the MUSAN dataset [32] and reverberated with [33].

Real recordings are represented by CALLHOME (CH) [34] and DIHARD II [35] datasets. Table III presents the details of the datasets used in the performed experiments. CALLHOME is a dataset based the telephone conversations. In the evaluation, we use the train and test split, the same as in [9]. The train part is used to fine-tune the systems trained using the simulated mixtures, and then it is evaluated on the test subset. For the experiment with a fixed number of speakers, only the subset of recordings that contains 2 speakers is used. DIHARD II is dataset used in the DIHARD challenge. It includes recordings from various domains. The development set is used for model fine-tuning, whereby the eval part is used for the evaluation. As the DIHARD II recordings are 16kHz, we downsample them to 8kHz in order to match our training sampling rate.

B. System Framework and Evaluation Measures

The system framework is implemented using the PyTorch library, with the use of available online implementation.[1] In our experiments we also present the results for the EEND-EDA official, Chainer implementation.[2] Our models are all trained with the Adam optimizer and a minibatch size of 64. For the initial training, we use the Noam learning rate scheduler as in [30], with 100 000 warm-up steps. For experiments with a constant number of speakers, we train them for 100 epochs, and for the ones with a variable number of speakers, we train them for 25 epochs. The fine-tuning is performed for 100 epochs, at a constant learning rate of 10^{-5}.

The final model is the average of models from the last 10 epochs. Only in the 2-speaker experiment in the simulated scenario, for the proposed systems, the 10 consecutive epochs are selected based on the validation loss value, which was required in order to avoid system overfitting. Models from 2 speaker experiments are used in experiments as weight initialization for adequate multi-speaker networks. The only exception is the EEND-NAA-Overest system, which is initialized by the EEND-NAA-2step 2-speaker trained model.

Parameter K is set individually for each experiment condition. In theory, K can be any arbitrary value smaller than the number of samples and larger than the maximum number of speakers. However, the higher K, the higher the training cost. Thus, we found that for the EEND-NAA-2step, it is reasonable to set it at least two times higher than the assumed maximum number of speakers, as K also indicates the number of samples for the second clustering. For EEND-NAA-Overest the number of speakers is dependent on the decision of the linear layer, thus we set K slightly higher than the assumed maximum number of speakers. An advantage of EEND-NAA, compared to other end-to-end diarization methods, is that K can be increased at inference time to accommodate a number of speakers larger than those seen in training.

All neural networks and each condition are trained with the $\lambda = 0.1$ parameter used in the loss equation (given by (8) and (10), respectively). Following the recipe in the official EEND-EDA repository, we used $\alpha = 1.0$, except for fine-tuning in the multi-speaker scenario where $\alpha = 0.1$ was used.

For all systems, the encoder is built of four transformer encoder layers, with 4-head attention mechanisms, 2048 dimensions in feed-forward layers, returning 256-dimensional embeddings. Single decoder blocks are always composed of two transformer decoder layers, following the same parameter setup as in the encoder part.

As input features, 23-dimensional log-Mel-filterbank coefficients are used, stacked with the context of 7 and a subsampling value of 10 [9], both for training and inference. In case of DIHARD II, at inference, subsampling of 5 is used. Moreover, following the original setup of the baseline EEND-EDA system, we also apply feature shuffling.

Following [9] we use chunk size of 500 frames, which corresponds to the segment size of 50s. For fine-tuning of DIHARD II, we use chunks of size 2000. During inference time, the entire recording is fed to the model. Due to such a processing, similar as in [9], the system has a limited audio length that it can process effectively. However, this limitation is a common challenge

[1] [Online]. Available: https://github.com/Xflick/EEND_PyTorch

[2] [Online]. Available: https://github.com/hitachi-speech/EEND

TABLE IV

DER Results for Models Trained With 2 Speaker Recordings

	Overlap, %			
System	35.3	28.0	19.6	CH
EEND-EDA [9]	2.69	2.44	2.60	8.07
EEND-EDA, Chainer	3.29	2.89	2.95	8.25
EEND-EDA, PyTroch	3.30	2.87	3.15	7.98
EEND-NAA-Fixed, $I = 1$	3.89	3.59	3.37	8.22
EEND-NAA-Fixed, $I = 3$	2.97	2.62	3.19	7.56
EEND-NAA-Overest	6.94	6.60	6.22	9.53
EEND-NAA-2step	3.02	2.91	3.52	7.87
EEND-NAA-1step	2.93	2.77	3.60	8.16

The test sets include three sets of simulated mixtures with different levels of speech overlap in % and the CALLHOME (CH) 2-speaker subset.

for the family of EEND systems and solving this issue is not addressed in this article.

In evaluations, for the simulated and CH datasets, system performance is measured using the Diarization Error Rate (DER) metric, with a 0.25 s collar. In case of the DIHARD II evaluation, we also include Jaccard Error Rate (JER) and set the collar tolerance as 0.00 s.

VII. Experimental Results and Discussion

In Section VII-A, we present the results and analysis of models trained with the recordings constrained to only two speakers. The 2-speaker models are then used as initialization for multi-speaker systems, which are evaluated in Section VII-B. In Section VII-C we further evaluate multi-speaker systems for the real recordings scenario.

A. *Evaluation on 2-Speaker Recordings*

Table IV presents the results for three different systems in the 2-speaker scenario experiment. The first three rows represent the EEND-EDA baseline, where the first row shows the results presented in the original paper [9], the second row shows our run of the EEND-EDA baseline as the official Chainer implementation, while the third presents our run as the PyTorch implementation. The results obtained with ours (PyTorch) and Chainer implementations are slightly worse than the ones presented in the original paper. It is difficult to indicate the reason for the discrepancy between the referenced and our results, which could arise from the mismatched running setups (e.g., one GPU versus multiple GPUs) or some differences in the generated mixtures. However, our runs allow us to fairly compare the different systems proposed in this article, regardless of the differences between the experimental setups of [9] and our experiments.

The next two rows present the EEND-NAA-Fixed system [13], with (i.e., $I = 3$) and without (i.e., $I = 1$) applying the iterative refinement. As described in [13], the EEND-NAA-Fixed results without iterative refinement for the simulated condition are slightly worse than those obtained with the EEND-EDA models. As can be observed, the application of the iterative refinement allows us to decrease DER metric values for all simulated and real conditions, achieving better results than the baseline EEND-EDA system.

The last three rows present the results of the EEND-NAA-Overest, EEND-NAA-2step, and EEND-NAA-1step systems proposed in this work. Note that the EEND-NAA-Overest is trained with parameter $K = 3$, while the EEND-NAA-2step is trained with $K = 4$ for the first clustering. Note that if the EEND-NAA-Overest and EEND-NAA-2step were both trained with the true (”perfect”) parameters $K = S = 2$, they would be the same as the EEND-NAA-1step model. Nevertheless, the higher value of K is selected to teach the expected behavior of the first decoder, which is to refine the cluster centers to discriminative mergeable representations for the EEND-NAA-2step or to pre-train the classification layer that estimates the number of speakers for the EEND-NAA-Overest. In the training phase for the EEND-NAA-1step system, parameter K is always equal to the number of speakers S in the recording. Thus, $K = 2$ for this experiment. For simulated recordings, both the EEND-NAA-2step and EEND-NAA-1step systems improve the set with 35.3% speech overlap, get comparable results for 28.0% overlap, and slightly degrade for 19.6% overlap. For the CALLHOME dataset, we can observe improvement for the EEND-NAA-2step system and slight degradation for EEND-NAA-1step over the baseline model. The DER values for EEND-NAA-Overest are at a satisfactory level. However, at the same time, the model represents the worst performance among all systems.

TABLE V

DER Results for Siumulated Test Recordings for Oracle and Estimated Number of Speakers

System	#1	#2	#3	#4	Avg
Oracle					
EEND-EDA [9]	0.16	4.26	8.63	13.31	6.59
EEND-EDA, Chainer	0.28	5.69	11.63	15.51	8.28
EEND-EDA, PyTorch	0.27	5.64	11.66	15.59	8.29
EEND-NAA-Fixed, $I = 1$	0.14	4.59	11.93	19.23	8.97
EEND-NAA-Fixed, $I = 3$	0.07	2.38	5.52	8.14	4.03
EEND-NAA-Overest	0.15	4.25	9.12	11.56	6.27
EEND-NAA-2step	0.08	2.26	5.97	10.99	4.83
EEND-NAA-1step	0.07	2.94	7.41	10.97	5.35
Estimated					
EEND-EDA [9]	0.39	4.33	8.94	13.76	6.86
EEND-EDA, Chainer	0.76	6.61	12.52	17.00	9.22
EEND-EDA, PyTorch	0.46	6.07	12.30	15.75	8.65
EEND-NAA-Overest	0.15	4.25	9.12	12.84	8.28
EEND-NAA-2step	0.08	4.31	6.83	12.26	5.87
EEND-NAA-1step	1.01	5.08	8.76	14.78	7.40

Test sets results includes sets of recordings with 1, 2, 3 and 4 speakers, and average among all test recordings.

B. *Evaluation on Simulated Multi-Speaker Recordings*

Table V presents the results for simulated multi-speaker recordings, both with a known (oracle) and an unknown (estimated) number of speakers. Firstly, let us analyze the outcome of the experiments presented in the first part of Table V for the simulated mixtures and an oracle number of speakers scenario. For EEND-NAA-Overest, $K = 6$ is used, while for EEND-NAA-2step, $K = 10$ is set. The choice of K is a compromise between the function of the first decoder and the computational load. Unlike the EEND-NAA-Overest system, the EEND-NAA-2step merges representations after the first decoder refinement, thereby aiming to have at least two potential centers per speaker. The EEND-NAA-1step system does not overestimate the number of clusters. Thus, the number of clusters is always set to $K =$

TABLE VI
DER Results for the Simulated Test set for the 1, 2, 3, and 4 Speakers in the Recording

SSAD condition	# spk condition	#1	#2	#3	#4
Fully est	oracle	0.08	2.26	5.97	10.99
	est	0.08	4.31	6.83	12.26
Oracle silence	oracle	0.07	2.25	5.75	8.80
	est	0.07	4.22	6.54	10.03
Oracle overlap	oracle	0.08	1.73	5.56	10.81
	est	0.08	3.57	6.48	11.85
Fully oracle	oracle	0.07	1.78	5.33	8.65
	est	0.07	3.55	6.17	9.62

The performance is presented for different SSAD condition (i.e. when the SSAD decision is fully estimated (est) by the model, fully oracle, or only overlap or silence is oracle) and for the known and unknown number of speakers. Presented results are for the EEND-NAA-2step system.

TABLE VII
Accuracy of the Overlap, Silence and Single-Speaker Region Detection for the Simulated Test set for the 1, 2, 3, and 4 Speakers in the Recording

	#1	#2	#3	#4
Overlap	100.00	95.58	96.11	96.74
Silence	98.90	98.81	98.35	96.44
SSAD	98.90	94.41	94.49	93.22

Presented results are for the EEND-NAA-2step system.

TABLE VIII
DER Results on CALLHOME for Oracle and Estimated Number of Speakers

System	#2	#3	#4	#5	#6	#All
Oracle						
EEND-EDA [9]	8.35	13.20	21.71	33.00	41.07	15.43
EEND-EDA, Chainer	8.95	14.01	22.38	30.10	39.32	15.83
EEND-EDA, PyTorch	8.93	14.03	21.38	31.92	42.70	16.13
EEND-NAA-Fixed, $I = 1$	8.11	16.98	24.32	39.13	42.05	17.97
EEND-NAA-Fixed, $I = 3$	6.54	12.91	17.61	28.59	38.97	13.76
EEND-NAA-Overest	7.59	12.73	19.59	29.69	48.12	15.29
EEND-NAA-2step	6.46	15.02	26.46	36.67	50.15	17.98
EEND-NAA-1step	7.50	13.11	19.54	30.90	36.71	14.46
Estimated						
EEND-EDA [9]	8.50	13.24	21.46	33.16	40.29	15.29
EEND-EDA, Chainer	9.01	13.89	21.83	29.69	40.42	15.69
EEND-EDA, PyTorch	8.49	14.37	21.45	31.80	44.21	15.93
EEND-NAA-Overest	9.16	14.84	20.21	29.17	39.56	15.90
EEND-NAA-2step	6.61	13.01	21.28	32.78	38.20	14.54
EEND-NAA-1step	8.64	12.74	19.45	30.92	37.16	14.73

Presented results are for each speaker number condition and for the whole test subset.

TABLE IX
Number of Utterances for Each Speaker Number for the CALLHOME Test set

#Speakers	2	3	4	5	6	Total
#Recordings	148	74	20	5	3	250

S during training. The very first system, the basic EEND-NAA-Fixed, presents an improvement over PyTorch EEND-EDA for the 1- and 2-speaker recordings and degradation over the 3- and 4-speaker sets, while the version with iterative refinement provides a clear gain, with 39% average relative improvement for all test sets compared to the EEND-EDA [9] baseline. The results of the EEND-NAA-1step are slightly worse than the basic EEND-NAA-Fixed with $I = 3$. However, its average performance over all test sets has a 35% relative improvement over the baseline. In turn, the EEND-NAA-2step performs comparably to the basic EEND-NAA-Fixed with only slight degradation for recordings with four speakers. The EEND-NAA-Overest system presents less impressive results. Nevertheless, it still improves over the baseline with 24% for the average value over all test sets.

The second part of Table V introduces the performances of the described systems when the number of speakers is unknown. Note that in this part, basic EEND-NAA-Fixed models are not included, as they lack the feature of estimating the number of speakers. For that scenario, in almost all cases the proposed systems achieve better results than EEND-EDA. The results of EEND-NAA-1step and EEND-NAA-2step vary to a greater degree compared to the known number of speakers case, which can be caused by the difference when the estimation of speaker number is applied in the models. In the EEND-NAA-1step system, the estimation is based on the frame-level embeddings filtered by the SSAD block, while the EEND-NAA-2step estimates with the refined cluster centers. Therefore, any contamination introduced by the EEND encoder or SSAD block directly affects the clustering decision for the EEND-NAA-1step.

Table VI presents the impact of the silence and overlap errors on the DER performance on the example of the EEND-NAA-2step system. The presented results are for both types of scenarios, with the known and unknown number of speakers. We distinguish four different SSAD conditions: (i) fully estimated, where SSAD decision is derived from the model, (ii) oracle overlap, where the SSAD overlap detection is replaced by the ground truth information, (iii) oracle silence, where the SSAD silence detection is replaced by the ground truth information, and (iv) fully oracle, where detection of both overlap and silence are derived from the ground truth labels. Firstly, let us note that the full oracle SSAD decision improves the results for all test sets. The most improvement we can observe for 2- and 4-speaker test sets. Comparing the results when we provide either only silence or only overlap oracle information, we can observe that proper overlap detection brings more improvement than silence one. However, it is correlated with the accuracy of the overlap and silence detection, which is presented in Table VII. As we can observe, the detection of the overlap is lower than silence detection for all multi-speaker recordings. Even though, we can consider the obtained accuracy as satisfactory, since all of them are way above 90%, we can notice that they may impact DER performance even up to 24% of the relative difference.

C. Evaluation on Real Multi-Speaker Recordings

This section presents the results of the systems for real multi-speaker recordings including CALLHOME telephone conversations, which is often used as benchmark in the literature, and DIHARD II which is a mixture of real recordings from various domains.

Table VIII presents the results for real multi-speaker recordings, both with a known (oracle) and an unknown (estimated) number of speakers. Before the analysis of the results, let us take into consideration the statistics of the CALLHOME test set. Table IX presents the number of utterances for each speaker condition. The higher the number of speakers, the fewer representative recordings are present in the test set, with only

TABLE X
RESULTS OF THE SELECTED STATE-OF-THE-ART SYSTEMS ON THE CALLHOME DATASET

Model	#Spk	#2	#3	#4	#5	#6	#All
X-vector [9]	Oracle	8.93	19.01	24.48	32.14	34.95	18.98
	Est	15.45	18.01	22.68	31.40	34.27	19.43
EEND-EDA [9]	Oracle	8.35	13.20	21.71	33.00	41.07	15.43
	Est	8.50	13.24	21.46	33.16	40.29	15.29
AED-EEND [12]	Oracle	6.79	12.36	19.84	34.42	37.08	14.00
	Est	6.96	12.56	18.26	34.32	44.52	14.22
EEND-VC [14]	Oracle	8.08	11.27	15.01	23.14	26.56	12.22
	Est	7.96	11.93	16.38	21.21	23.10	12.49
EEND-GLA [25]	Oracle	-	-	-	-	-	-
	Est	6.94	11.42	14.49	29.76	24.09	11.92

three utterances resulting in the 6-speaker condition. As a consequence, the results for the low-numbered subsets may be less reliable and present ambiguous conclusions. In addition, note that as mentioned in Section VI, all systems have been fine-tuned on the CALLHOME train subset, which includes up to 7 speakers. Thus, in the fine-tuning conditions, the system parameter has been modified to $K = 8$ for EEND-NAA-Overest.

In the oracle part of Table VIII, the best results are obtained by the EEND-NAA-Fixed system with iterative refinement. From the newly proposed systems, the best performance can be observed for the EEND-NAA-1step model, with only slightly worse results than the basic EEND-NAA-Fixed system version. The EEND-NAA-2step system shows improvement over the baseline for subset with 2 speakers, with the most recordings from all subsets. Contrary to the previous experiment EEND-NAA-Overest improves compared to the baseline.

For the estimated speaker number condition, in Table VIII, the EEND-NAA-2step system shows the lowest result for the whole CALLHOME set. For the higher speaker number, the system tends to underestimate the speaker number, resulting in a lower DER value. The diarization results of EEND-NAA-1step are slightly worse than those of the EEND-NAA-2step system. Nonetheless, EEND-NAA-1step still provides an 8% relative improvement for the whole set. The EEND-NAA-Overest exhibits performance that is similar to the oracle number of speakers condition. In all cases, except in the 6-speaker set, it provides comparable results to those of the baseline.

In this section, we also present Table X with the results of the most relevant for our work reference systems on the CALLHOME dataset. X-vector model [9] represents the standard cluster-based method which uses x-vectors as speaker embeddings, EEND-EDA is our reference model from [9], while AED-EEND [12], EEND-VC [14], and EEND-GLA [25] denote other three state-of-the-art systems introduced in Section II. We would like to point out that in case of the systems presented in the fourth and fifth row of Table X, the training procedure (i.e. parameters used to create simulated mixtures and the number of epochs applied) is different than in our evaluation protocol which has a large impact on the final results. Thus, we do not compare directly these results with the results from our experiments but present them only as a frame of reference.

For further comprehensive evaluation with real data, we also evaluated the selected systems on the DIHARD II dataset. The results of this experiment are presented in Table XI. For the DIHARD II dataset, we compare the baseline EEND-EDA and EEND-NAA-Fixed systems with the proposed EEND-NAA-2step system. In particular, we present EEND-NAA-2step system for three configurations of parameter K: (i) trained with $K_{\text{train}} = 10$ and evaluated with $K_{\text{infer}} = 10$, (ii) trained with $K_{\text{train}} = 10$ and evaluated with $K_{\text{infer}} = 20$, and (iii) trained with $K_{\text{train}} = 10$ and evaluated with $K_{\text{infer}} = 20$. In the first row of Table XI, as a reference, we also provide the EEND-EDA result from [9]. As can be observed, both NAA-based systems achieve slightly better results than EEND-EDA. In case of the EEND-NAA-Fixed the presented performance is lower than the literature version of the EEND-EDA, especially in terms of JER metric. In addition, for the EEND-NAA-2step system, we can observe that the results are nearly the same between each other, regardless of the K value used in training and inference. This supports the statement that parameter K does not limit the models capacity and can be changed at inference with respect its initial value set in training.

TABLE XI
DER AND JER RESULTS ON DIHARD II DATASET

Model	DER	JER
EEND-EDA [9]	32.59	55.99
EEND-EDA	34.35	57.38
EEND-NAA-Fixed, $I = 3$	33.95	49.54
EEND-NAA-2step, $K_{\text{train}} = 10$, $K_{\text{infer}} = 10$	29.66	52.27
EEND-NAA-2step, $K_{\text{train}} = 10$, $K_{\text{infer}} = 20$	30.18	53.04
EEND-NAA-2step, $K_{\text{train}} = 20$, $K_{\text{infer}} = 20$	29.49	52.30

VIII. Conclusion

This article has proposed a new development of the EEND diarization system with non-autoregressive attractors (EEND-NAA) that is capable of working under the condition of a variable and unknown number of speakers. In particular, we presented three new systems, which in this article are referred to as EEND-NAA-Overest, EEND-NAA-1step, and EEND-NAA-2step, respectively, that follow the EEND-NAA framework introduced by the authors in a former conference paper, in which non-autoregressive attractor estimation is integrated into the end-to-end pipeline. The back-end structure has been extended to two decoders that each refine the initial attractors to the desired representations. As indicated by the results of the performed experiments, in general, the proposed systems outperform the baseline and even have the potential to further decrease the diarization error rate (DER) value.

For conditions with a fixed number of speakers, the best results were obtained with the EEND-NAA-Fixed and EEND-NAA-2step systems. While in the condition with a fixed number of speakers (Section VII-A), the results presented by the proposed systems are competitive to the baseline models presented in the literature (i.e., the baseline EEND-EDA and our EEND-NAA), we can observe a clear gain for the condition of a variable number of speakers. Both EEND-NAA-2step and EEN-NAA-1step achieve up to 42% relative improvement over the baseline EEND-EDA for all simulated recordings for the known number of speakers and 32% for the estimated number of speakers. Although one of the models presented in this article, which is referred to as EEND-NAA-Overest, did not achieve as low DER values as the other two proposed systems for the 2-speaker experiment, it also achieved good improvement over the baseline EEND-EDA in scenarios with more than two speakers.

The performance of the proposed systems has been confirmed by experiments performed on the CALLHOME dataset that contains the recordings from real-life telephone conversations. During the analysis of the results, the main conclusions were primarily drawn based on the performance on the sets that contain 2 and 3 speakers, as well as for the entire test set. The motivation for using these subsets in the analysis is that their results are the most reliable since the subsets containing from 4 to 6 speakers have at most only 20 utterances. The evaluation also included the DIHARD II dataset, where we selected the most important structures from the previous experiments. We presented the improvement over EEND-EDA system and proved that parameter K does not limit the EEND-NAA performance.

Acknowledgment

The authors gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Centers: ACK Cyfronet AGH) for providing computer facilities and support within computational Grants PLG/2022/015954 and PLG/2023/016823.

References

[1] S. H. Shum, N. Dehak, R. Dehak, and J. R. Glass, "Unsupervised methods for speaker diarization: An integrated and iterative approach," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 10, pp. 2015–2028, Oct. 2013.

[2] D. Garcia-Romero, D. Snyder, G. Sell, D. Povey, and A. McCree, "Speaker diarization using deep neural network embeddings," in *Proc. 2017 IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 4930–4934.

[3] M. Diez, L. Burget, S. Wang, J. Rohdin, and J. Černocký, "Bayesian HMM based X-vector clustering for speaker diarization," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2019, pp. 346–350.

[4] L. Bullock, H. Bredin, and L. P. García-Perera, "Overlap-aware diarization: Resegmentation using neural end-to-end overlapped speech detection," in *Proc. ICASSP 2020-2020 IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 7114–7118.

[5] D. Raj, Z. Huang, and S. Khudanpur, "Multi-class spectral clustering with overlaps for speaker diarization," in *Proc. 2021 IEEE Spoken Lang. Technol. Workshop*, 2020, pp. 582–589.

[6] H. Bredin and A. Laurent, "End-to-end speaker segmentation for overlap-aware resegmentation," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 3111–3115.

[7] X. Fang et al., "A deep analysis of speech separation guided diarization under realistic conditions," in *Proc. IEEE 2021 Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, 2021, pp. 667–671.

[8] Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with permutation-free objectives," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2019, pp. 4300–4304.

[9] S. Horiguchi, Y. Fujita, S. Watanabe, Y. Xue, and K. Nagamatsu, "End-to-end speaker diarization for an unknown number of speakers with encoder-decoder based attractors," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 269–273.

[10] K. Kinoshita, M. Delcroix, and N. Tawara, "Integrating end-to-end neural and clustering-based diarization: Getting the best of both worlds," in *Proc. ICASSP 2021 IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 7198–7202.

[11] D. Wang, X. Xiao, N. Kanda, T. Yoshioka, and J. Wu, "Target speaker voice activity detection with transformers and its integration with end-to-end neural diarization," in *Proc. ICASSP 2023 IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2023, pp. 1–5.

[12] Z. Chen, B. Han, S. Wang, and Y. Qian, "Attention-based encoder-decoder network for end-to-end neural speaker diarization with target speaker attractor," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2023, pp. 3552–3556.

[13] M. Rybicka, J. Villalba, N. Dehak, and K. Kowalczyk, "End-to-end neural speaker diarization with an iterative refinement of non-autoregressive attention-based attractors," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2022, pp. 5090–5094.

[14] K. Kinoshita, M. Delcroix, and N. Tawara, "Advances in integration of end-to-end neural and clustering-based diarization for real conversational speech," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 3565–3569.

[15] K. Kinoshita, M. Delcroix, and T. Iwata, "Tight integration of neural- and clustering-based diarization through deep unfolding of infinite Gaussian mixture model," in *Proc. ICASSP 2022-2022 IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 8382–8386.

[16] I. Medennikov et al., "Target-speaker voice activity detection: A novel approach for multi-speaker diarization in a dinner party scenario," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 274–278, 2020.

[17] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 788–798, May 2011.

[18] N. Chen, S. Watanabe, J. Villalba, P. Żelasko, and N. Dehak, "Non-autoregressive transformer for speech recognition," *IEEE Signal Process. Lett.*, vol. 28, pp. 121–125, 2021.

[19] E. G. Ng, C.-C. Chiu, Y. Zhang, and W. Chan, "Pushing the limits of non-autoregressive speech recognition," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 3725–3729, 2021.

[20] Y. Higuchi, H. Inaguma, S. Watanabe, T. Ogawa, and T. Kobayashi, "Improved mask-CTC for non-autoregressive end-to-end ASR," in *Proc. ICASSP 2021-2021 IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 8363–8367.

[21] W. Chan, C. Saharia, G. Hinton, M. Norouzi, and N. Jaitly, "Imputer: Sequence modelling via imputation and dynamic programming," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1403–1413.

[22] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *Proc. 2017 IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 241–245.

[23] Y. Fujita, N. Kanda, S. Horiguchi, Y. Xue, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with self-attention," in *Proc. 2019 IEEE Autom. Speech Recognit. Understanding Workshop*, 2019, pp. 296–303.

[24] Y. Fujita, S. Watanabe, S. Horiguchi, Y. Xue, and K. Nagamatsu, "End-to-end neural diarization: Reformulating speaker diarization as simple multi-label classification," 2020, *arXiv:2003.02966*.

[25] S. Horiguchi, S. Watanabe, P. García, Y. Xue, Y. Takashima, and Y. Kawaguchi, "Towards neural diarization for unlimited numbers of speakers using global and local attractors," in *Proc. 2021 IEEE Autom. Speech Recognit. Understanding Workshop*, 2021, pp. 98–105.

[26] S. Horiguchi, Y. Fujita, S. Watanabe, Y. Xue, and P. Garcia, "Encoder-decoder based attractors for end-to-end neural diarization," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 30, pp. 1493–1507, 2022.

[27] C. Yang and Y. Wang, "Robust end-to-end speaker diarization with generic neural clustering," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2022, pp. 1471–1475.

[28] K. Kinoshita, T. von Neumann, M. Delcroix, C. Boeddeker, and R. Haeb-Umbach, "Utterance-by-utterance overlap-aware neural diarization with Graph-PIT," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2022, pp. 1486–1490.

[29] Y. Fujita, T. Komatsu, R. Scheibler, Y. Kida, and T. Ogawa, "Neural diarization with non-autoregressive intermediate attractors," in *Proc. ICASSP 2023 IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2023, pp. 1–5.

[30] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.

[31] O. Thiergart, M. Taseska, and E. A. P. Habets, "An informed parametric spatial filter based on instantaneous direction-of-arrival estimates," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 12, pp. 2182–2196, Dec. 2014.

[32] D. Snyder, G. Chen, and D. Povey, "MUSAN: A music, speech, and noise corpus," 2015, *arXiv:1510.08484v1*.

[33] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *Proc. 2017 IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 5220–5224.

[34] M. Przybocki and A. Martin, *2000 NIST Speaker Recognition Evaluation LDC2001S97. Web Download.* Philadelphia, PA, USA: Linguistic Data Consortium, 2001.

[35] N. Ryant et al., "The second DIHARD diarization challenge: Dataset, task, and baselines," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.* 2019, pp. 978–982.

Magdalena Rybicka received the B.Eng. and M.Sc. degrees in electronics and telecommunications, in 2018 and 2019, respectively, from the AGH University of Krakow, Kraków, Poland, where she is currently working toward the Ph.D. degree with the Institute of Electronics. From 2022 to 2024, she was a visting Ph. D. Student with the Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD, USA. She is also a Research Assistant with the Institute of Electronics, AGH University of Krakow. Her research interests include speaker diarization, verification and speech separation. She was the recipient of the Fulbright Junior Research Award.

Jesús Villalba (Member, IEEE) received the M.S. degree in telecommunications engineering and the Ph.D. degree in biomedical engineering from the University of Zaragoza, Zaragoza, Spain, in 2004 and 2014, respectively. His thesis focused on several topics related to speaker recognition in adverse environments. In 2016, he joined the Johns Hopkins Center for Language and Speech Processing, Baltimore, MD, USA, as a Postdoctoral Fellow. He was appointed as an Assistant Research Professor in 2019. He is currently an Assistant Research Professor with the Department of Electrical and Computer Engineering and an affiliate of the Center for Language and Speech Processing. His research interests include information extraction from speech, such as speaker identity, language, age, and emotion, speaker diarization, and unsupervised learning for speech-related applications.

Thomas Thebaud (Member, IEEE) received the Ph.D. degree in biometric antispoofing from Le Mans University, Le Mans, France. He was with Orange Labs, Cesson-Sévigné, France, and completed his Postdoctoral research with the Center for Language and Speech Processing (CLSP), Johns Hopkins University, Baltimore, MD, USA. He is currently an Assistant Research Scientist with CLSP. His research interests include emotion recognition, handwriting analysis, and summarization.

Najim Dehak (Senior Member, IEEE) received the Ph.D. from the School of Advanced Technology, Montreal, QC, Canada, in 2009. He was with the Computer Research Institute of Montreal, Montreal. He is well known as a leading developer of the I-vector representation for speaker recognition. He first introduced this method, which has become the state-of-the-art in this field, during the 2008 summer Center for Language and Speech Processing workshop at Johns Hopkins University. This approach has become one of most known speech representations in the entire speech community. He was a Research Scientist in the Spoken Language Systems Group with the MIT Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA. He is currently a Faculty member with the Department of Electrical and Computer Engineering, Johns Hopkins University. His research interests include machine learning approaches applied to speech processing, audio classification, and health applications. He is a Member of the IEEE Speech and Language Technical Committee.

Konrad Kowalczyk (Senior Member, IEEE) received the B.Eng. and M.Sc. degrees in telecommunication from the AGH University of Krakow, Kraków, Poland, in 2005, the Ph.D. degree in electronics from Queen's University, Belfast, U.K., in 2009, and the Habilitation (D.Sc.) degree in information and communication technology from the AGH University of Krakow in 2020. He was a Visiting Scholar with Stanford University, Stanford, CA, USA, in 2007, University of York, York, U.K., in 2008, and Aalto University, Espoo, Finland, in 2016. From 2009 to 2011, he was a Postdoctoral Research Fellow with the University of Erlangen-Nuremberg, Erlangen, Germany. From 2012 to 2014, he was an Associate Researcher with the Fraunhofer Institute for Integrated Circuits (IIS), Erlangen, and International Audio Laboratories Erlangen, Erlangen. In 2015, he joined AGH University of Krakow, where he is currently an Associate Professor and Head of AGH's Signal Processing Group. He was a finalist of the IEEE Best Student Paper Contest at ICASSP 2007. He was the recipient of the AES Student Technical Paper Award at the AES Convention in 2008, and co-author with the Best Student Paper Award at IWAENC 2014. He is a Member of the Technical Area Committee on Acoustic, Speech and Music Signal Processing of the European Association for Signal Processing (EURASIP). He was a Guest Editor of *EURASIP Journal on Audio, Speech, and Music Processing*. Since 2022, he has been an Associate Editor for the IEEE SIGNAL PROCESSING LETTERS.

V Joint Diarization and Separation Using SepFormer with Non-Autoregressive Attractors

M. Rybicka, K. Kowalczyk, T. Thebaud, N. Dehak, J. Villalba "*Joint Diarization and Separation Using SepFormer with Non-Autoregressive Attractors*", IEEE Signal Processing Letters, 2025.

Joint Diarization and Separation Using SepFormer With Non-Autoregressive Attractors

Magdalena Rybicka, Konrad Kowalczyk, *Senior Member, IEEE*, Thomas Thebaud, *Member, IEEE*, Najim Dehak, *Senior Member, IEEE*, and Jesús Villalba, *Member, IEEE*

***Abstract*—Speaker diarization and speech separation both aim to track speaker activity in multi-speaker recordings, but they differ in their granularity. Diarization provides a binary indication of whether a speaker is active within a given time frame, whereas speech separation produces individual audio signals, each containing the isolated speech of a specific speaker. Recently, there has been growing interest in approaches that unify diarization and speech separation, particularly those leveraging neural models trained jointly to enhance performance in both tasks. In this letter, we propose a single neural model for joint speaker diarization and speech separation. Our model estimates speaker representations using a non-autoregressive attractor generation mechanism integrated into a modified SepFormer model. We present two variants of the model, designed for scenarios with sparse or highly overlapping speech, which achieve relative improvements of 51% for both separation and diarization over state-of-the-art methods, as evaluated on the LibriMix, LibriheavyMix and CALLHOME datasets.**

***Index Terms*—Attractor mechanism, clustering, end-to-end, non-autoregressive model, speaker diarization, speech separation.**

I. Introduction

Diarization answers the question “who spoke when” by predicting the time stamps during which each speaker is active in the recording. In turn, separation estimates the individual waveforms of each speaker, typically by applying time-frequency speech activity masks. In fact, separation and diarization are closely related and can be seen as mutually complementary tasks. Hence, a unified model has the potential to boost performance and help address the challenges inherent in each task. Speech separation simplifies the diarization task to voice activity detection (VAD) applied on each separated audio track, while the time stamps of segments that contain speech of a particular speaker could guide the separation process. A combination of independent modules for diarization and separation is presented in [1], [2]. A method for speaker separation via neural diarization is introduced in [3], where End-to-End Neural Diarization with Encoder-Decoder based Attractors (EEND-EDA) [4] is used to produce speaker representations and count the speaker number. The representations are composed into a two-channel embedding sequence, which is fed as an additional input to the separator to help track specific speakers. A similar approach is taken in [5], where speech separation is performed for a flexible number of speakers using the SepEDA structure built on a SepFormer [6] with autoregressive attractors and the speaker count provided by the EDA block. Recently, there has been growing interest in the joint modeling of diarization and separation. In [7], the Joint End-to-End Neural Speaker Diarization and Separation (EEND-SS) is introduced, combining ConvTasNet [8] for separation with EEND-EDA [4] for diarization in a single, jointly trained structure. In Target-Speaker based Separation (TS-SEP) [9], Target-Speaker VAD (TS-VAD) based diarization [10] is used to retrain the binary time-activity result to output the time-frequency masks. At the same time, the diarization result is derived based on the values of the separation masks. Another approach PixIT [11] is based on the Dual-Path RNN (DPRNN) [12], with an additional diarization output added in parallel to the separation decoder and a combination of features from WavLM and the separation encoder.

In this article, we propose a neural model for the joint task of speech separation and speaker diarization. Similarly to SepEDA [5], our approach is based on a modified SepFormer model. However, unlike SepEDA, we enable true joint modeling by extending the separation network with a diarization component and employing an objective function that incorporates both diarization and separation losses. In our approach, the bottleneck features from the SepFormer-based separator are used to estimate speaker representations (the so-called attractors), which serve both for diarization and element-wise modulation of dual-path embeddings in the separation branch. We introduce two variants of the model, tailored for the recordings with either sparse or highly overlapping speech. For sparsely overlapping scenarios, which are typical in diarization tasks, we adopt a Cluster-based Attractor (CA) mechanism, originally introduced for the diarization task in [13], [14]. This method aggregates, detects, and clusters embeddings corresponding to single-speaker frames to compute attractors used to enhance separation, while diarization decisions are derived from the final

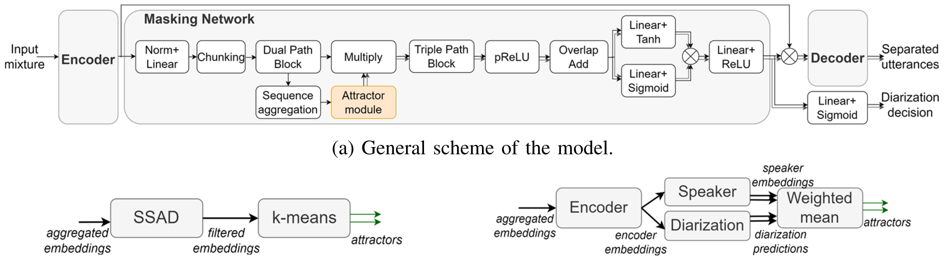
Received 30 May 2025; revised 14 July 2025; accepted 15 July 2025. Date of publication 18 July 2025; date of current version 29 July 2025. This work was supported in part by the National Science Centre, Poland, under Grant 2021/42/E/ST7/00452, in part by the AGH University of Krakow under the program “Excellence initiative – research university”, and in part by Polish high-performance computing infrastructure PLGrid (HPC Centers: ACK Cyfronet AGH) under Grant PLG/2023/016823 and Grant PLG/2024/017659. The associate editor coordinating the review of this article and approving it for publication was Dr. Xiao-Lei Zhang. *(Corresponding author: Magdalena Rybicka.)*

Magdalena Rybicka was with Johns Hopkins University, Baltimore, MD 21218 USA. She is now with the AGH University of Krakow, 30-059 Krakow, Poland (e-mail: mrybicka@agh.edu.pl).

Konrad Kowalczyk is with the AGH University of Krakow, 30-059 Krakow, Poland (e-mail: konrad.kowalczyk@agh.edu.pl).

Thomas Thebaud, Najim Dehak, and Jesús Villalba are with the CLSP Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: tthebau1@jhu.edu; ndehak3@jhu.edu; jvillal7@jhu.edu).

Digital Object Identifier 10.1109/LSP.2025.3590325



(a) General scheme of the model.

(b) Cluster-based Attractors (CA) estimation.

(c) Diarization-based Attractors (DA) estimation.

Fig. 1. Diagrams of the general structure of the joint separation and diarization model and attractor generation modules.

separation masks. For highly overlapping speech conditions–typical in separation tasks–we introduce Diarization-based Attractor (DA) estimation. Here, the aggregated embeddings are passed through an EEND diarization encoder [4] followed by speaker and diarization modules, to produce both diarization predictions and speaker embeddings. These embeddings are then used to compute the final attractors. Both variants were evaluated on the LibriMix, LibriheavyMix and CALLHOME datasets, showing improved performance over existing joint models, while at minimum maintaining the performance of state-of-the-art systems designed for the individual tasks. The baseline and the proposed model structures are described in Sections II and III. The experimental setup and results are presented in Sections IV and V; Section VI provides the summary.

II. Baseline System Overview

The baseline structure, adopted in this work, builds upon the SepFormer [6] and SepEDA [5] frameworks, both of which directly estimate separated speech waveforms. The SepFormer comprises three components: an *encoder*, *masking network*, and *decoder*. The raw waveform input $\mathbf{x} \in \mathbb{R}^{1\times T}$, of length T samples, enters the encoder consisting of a 1-D convolutional layer with 256 filters, kernel-size of 16, stride of 8, followed by a ReLU activation. As shown in Fig. 1(a), the resulting time-feature matrix $\mathbf{h} \in \mathbb{R}^{L\times D}$–where L denotes the subsampled time dimension and D the feature dimension–, is passed through the normalization and linear layers within the masking network. The output is then divided into overlapping chunks, producing $\mathbf{h}' \in \mathbb{R}^{C\times K\times D}$, where C is the number of chunks and K is the chunk size, set to 250. These segments are processed by the dual-path block, a standard component of the SepFormer architecture. In SepFormer [6] multiple dual-path blocks are used. Following the SepEDA modifications [5], our architecture employs one dual-path block and one triple-path block. The dual-path block consists of intra-chunk and inter-chunk transformer layers: the intra-chunk captures short-term dependencies within chunks, while the inter-chunk models long-term dependencies across chunks.

In SepEDA [5], the output of the dual-path block is passed to the aggregation module, whose task is to aggregate the representations at the chunk level $\mathbf{g} \in \mathbb{R}^{C\times D}$. The aggregation is done through self-attentive weighted subspace projection [5]. This representation is fed into the attractor module, described in Section III for the two proposed variants. The resulting attractors are then applied to the dual-path output via element-wise multiplication, yielding the representation $\mathbf{g}' \in \mathbb{R}^{C\times K\times D\times S}$, S – number of speakers. In SepEDA [5], separation-only model, autoregressive attractors from Encoder-Decoder-based Attractors (EDA) [4] are used. Next, the embeddings are processed by the triple-path block, which extends dual-path with an additional inter-channel transformer block that models inter-speaker dependencies. The output is processed with pReLU, accompanied by an overlap-add that merges the chunks into $\mathbf{h}''' \in \mathbb{R}^{L\times D\times S}$. This output is fed into two parallel linear layers—one with a *Tanh* and the other with a *Sigmoid* — whose outputs are combined via element-wise multiplication. Final speaker masks, $\mathbf{m} \in \mathbb{R}^{L\times D\times S}$, are obtained through another linear layer with ReLU activation.

Finally, the decoder–a transposed convolutional layer with the same parameters as the encoder–processes the masked encoder time-freq features $\mathbf{h}$ to reconstruct the separated speech waveforms.

III. Proposed Joint Separation and Diarization

This letter presents a novel approach for joint speech separation and diarization (SepDiar) that builds upon a separation architecture and incorporates non-autoregressive attractor generation mechanisms into the modified SepFormer model (Fig. 1(a)). We focus on two ways of producing the attractors–representations of the speakers present in the utterance. For sparsely overlapping speech recordings, typical for diarization tasks, we propose the SepDiarCA model, which employs non-autoregressive cluster-based attractor generation. For highly overlapping speech, more common for separation tasks, we propose the SepDiarDA model, which relies on diarization-based attractor generation. Unlike SepEDA [5], which employs autoregressive attractors from EDA [4] solely for the separation task, both SepDiar variants are trained jointly with combined diarization and separation objectives, and generate non-autoregressive attractors tailored for the joint task.

A. Cluster-Based Attractor Estimation (SepDiarCA)

The SepDiarCA model employs Cluster-based Attractor (CA) estimation, that is primarily based on time-frame embeddings associated with single speakers. This non-autoregressive cluster-based attractor generation mechanism, originally proposed for diarization in [13], [14], is particularly effective for sparsely overlapping speech such as in conversational scenarios. The diagram of CA is in Fig. 1(b). Firstly, the aggregated embeddings $\mathbf{g} \in \mathbb{R}^{C\times D}$ are processed by a Single Speaker Activity Detection (SSAD) module, which detects embeddings containing single-speaker speech and filters out those representing overlap or silence. SSAD consists of a single transformer encoder layer followed by one-output linear classification layer that decides whether each embedding corresponds to a single speaker or not. The resulting sequence $\mathbf{c} \in \mathbb{R}^{C' \times D}$, where C' is the number of single-speaker embeddings, is clustered by the k-means algorithm, with the number of clusters equal to the number of speakers present in the recording. The estimated cluster centers $\mathbf{c}' \in \mathbb{R}^{S \times D}$ are selected as attractors. These attractors are then applied to the dual-path embeddings via element-wise multiplication to enhance speech separation. The diarization predictions are retrieved from the separation masks predicted by the masking network. A linear layer with *Sigmoid* activation applied on top of all feature masks $\mathbf{m}$ for a particular speaker at a particular time frame, returns the $\mathbf{d} \in \mathbb{R}^{L \times S}$ diarization predictions.

For SepDiarCA model training, we minimize a sum of separation, diarization, and SSAD classification losses,

$$\mathcal{L}_{\text{SepDiarCA}} = \mathcal{L}_{\text{sep}} + \mathcal{L}_{\text{diar}} + \mathcal{L}_{\text{SSAD}} \ , \quad (1)$$

where $\mathcal{L}_{\text{sep}}$ is the Scale-Independent SNR (SI-SNR) [7], $\mathcal{L}_{\text{diar}}$ is the diarization binary cross-entropy loss, computed as in [4], while $\mathcal{L}_{\text{SSAD}}$ is a binary classification cross-entropy loss which decides whether a particular time frame corresponds to a single-speaker embedding or not.

B. Diarization-Based Attractor Estimation (SepDiarDA)

To effectively handle recordings with high speech overlap, we propose to incorporate Diarization-based Attractor (DA) estimation, which replaces the clustering used in CA. We refer to this model as SepDiarDA. The DA diagram is in Fig. 1(c). The processing flow is inspired [15] which is based on a classical EEND. In the proposed SepDiarDA, the aggregated embeddings $\mathbf{g} \in \mathbb{R}^{C \times D}$ are processed by the encoder module, which is built with two transformer encoder layers. Next, the resulting embedding sequence $\mathbf{e} \in \mathbf{R}^{C \times D}$ is processed in parallel by two branches, namely the diarization and the speaker modules. The diarization estimates speaker presence using a linear layer, with S_{max} (max. speaker number) outputs, where output $\mathbf{e}_d \in \mathbf{R}^{C \times S_{\text{max}}}$ indicates the probability that a particular speaker is present in a given embedding. Simultaneously, the speaker module–which consists of S_{max} linear layers–projects the encoder embeddings into S_{max} distinct speaker representation sequences, forming $\mathbf{e}_s \in \mathbb{R}^{C \times S_{\text{max}} \times D}$. The attractors are computed as a weighted average of speaker representations, using the diarization probabilities $\mathbf{e}_d$ as soft weights. Although SepDiarDA diarization results can be derived from the DA, we extract diarization decisions from the masks via linear layer, similarly to SepDiarCA, to achieve higher resolution, while DA acts as a precise attractor estimator. Nevertheless, when SepDiarDA is used solely for diarization, the structure up to the DA operation is sufficient, enabling a much smaller model.

The SepDiarDA can count the number of speakers by detecting the silent speakers in DA diarization decision, similar to [16]. We compute the mean diarization probability for each speaker and compare it to a threshold $\tau = 0.05$. If the mean is below τ, the corresponding speaker is discarded. The loss function for the SepDiarDA training is a sum of separation $\mathcal{L}_{\text{sep}}$, $\mathcal{L}_{\text{DA}}$ speaker activity, and mask diarization $\mathcal{L}_{\text{diar}}$ loss:

$$\mathcal{L}_{\text{SepDiarDA}} = \mathcal{L}_{\text{sep}} + \mathcal{L}_{\text{DA}} + \mathcal{L}_{\text{diar}} \ . \quad (2)$$

$\mathcal{L}_{\text{sep}}$ and $\mathcal{L}_{\text{diar}}$ are calculated in the same manner as for SepDiarCA. $\mathcal{L}_{\text{DA}}$ is the speaker-activity binary cross-entropy loss obtained in the DA attractor generation module and calculated in an analogous way as the diarization loss.

IV. Evaluation Setup

Experimental evaluation is using SparseLibri2Mix [17], CALLHOME (CH) [18], Libri2Mix, Libri3Mix [17] and LibriheavyMix [19]. Libri2Mix and Libri3Mix are created by mixing LibriSpeech [20] utterances and WHAM! [21] noise samples, representing highly overlapping speech scenarios. The train-clean-100, dev-clean, and test-clean sets are used for training, validation, and test. Mixtures are generated at 8 kHz sampling rate in *min mode*, where the mixture duration matches the shortest source utterance. SparseLibri2Mix simulates 2-speaker conversational speech with limited overlap. It is generated using the SparseLibriMix scripts[1]. The test set comprises six overlap conditions, each with 500 mixtures. Following [7], we generated 5000 mixtures per condition for training and validation with a 90/10 split. In our experiments, we use clean version of test sets since noise samples were of insufficient length to generate noisy sets for training and validation [17]. LibriheavyMix [19] is a recent large-scale dataset with simulated reverberant mixtures. The training portion includes 1- to 4-speaker recordings. To focus on multi-speaker separation, we select a subset from the *train-small* set that contains mixtures with 2 to 4 speakers. For validation, the *dev* subset is used. CH contains real-life telephone conversations and is a common benchmark for diarization. CH does not have the ground truth for separation; thus, we used single-speaker regions from the train part of CH and simulated 2-speaker mixtures to adapt models for the joint task. For the test, we used the 2-speaker subset of the test part [4].

For evaluation, we used SI-SDR improvement (SI-SDRi) [22] and Diarization Error Rate (DER) with 0.00 collar, except CH results, which used 0.25 s collar. In the experiment with a flexible number of speakers, the Speaker Counting Accuracy (SCA), expressed in %, is reported. We used Adam optimizer with a learning rate $1.5e^{-4}$, patience of 2 and batch size 1. To fit recordings into GPU memory, we limited lengths to under 25 s for 2–3 speakers and 20 s for 4 speakers in LibriheavyMix. Training was stopped if the validation loss does not improve for five consecutive epochs.

V. Experimental Results and Discussion

A. SparseLibri2Mix and CH for Conversational Recordings

Table I presents the results for the sparsely overlapped SparseLibri2Mix mixtures and real-life recordings from CH. Our proposed models, SepDiarCA and SepDiarDA, are compared with baselines: EEND-SS [7] joint diarization and separation, a diarization-only EEND-EDA [4], separation models SepFormer [6] and SepEDA [5]. We observe notable differences between the results reported in [7] and the ones obtained using our experimental setup. To ensure fair comparison, we include EEND-EDA (Ours), trained using setup described in Section IV. Since proposed models and SepEDA (∼13 M params) are twice smaller than SepFormer (∼26 M params), we present SepFormerS - SepFormer with half the number of the inter- and intra-chunk blocks. SepFormerS has comparable

[1] https://github.com/popcornell/SparseLibriMix

TABLE I
SPARSELIBRI2MIX AND CALLHOME (CH) RESULTS

Overlap, %	0		20		40		60		80		100		CH
System	SI-SDRi	DER	SI-SDRi	DER	SI-SDRi	DER	SI-SDRi	DER	SI-SDRi	DER	SI-SDRi	DER	DER
EEND-SS [7]*	22	4.5	9	6.5	7.5	5.4	7.5	5.2	7	4.6	6	3.5	–
EEND-EDA [7]*	–	13.0	–	12.0	–	10.3	–	8.0	–	6.6	–	5.2	–
EEND-EDA (Ours)	–	10.07	–	11.28	–	9.99	–	8.57	–	7.30	–	6.48	21.47
SepFormer	42.65	–	21.45	–	18.72	–	17.33	–	16.56	–	16.05	–	–
SepFormerS	39.61	–	21.61	–	18.90	–	17.65	–	16.73	–	16.18	–	–
SepEDA	40.76	–	20.30	–	17.45	–	16.37	–	15.61	–	15.00	–	–
SepFormerS + Diar	40.77	1.04	21.43	2.42	18.76	2.28	17.35	2.40	16.79	2.32	16.44	1.76	9.40
SepEDA + Diar	40.78	1.42	21.56	2.44	18.87	2.23	17.40	2.43	16.64	2.30	16.22	1.79	10.35
SepDiarCA	42.57	0.88	22.11	2.25	19.14	2.14	17.81	2.15	17.14	2.19	16.05	1.76	**6.80**
SepDiarDA	**42.93**	**0.59**	**22.32**	**2.15**	**19.67**	**2.00**	**18.33**	**1.99**	**17.54**	**2.14**	**17.16**	**1.48**	7.40

* indicates the results derived from the plot diagram in [7].

TABLE II
RESULTS ON LIBRIHEAVYMIX

	dev		test-clean		test-other	
System	SI-SDRi	DER	SI-SDRi	DER	SI-SDRi	DER
pyannote [19]	–	41.10	–	40.57	–	38.60
SepFormerS + Diar	6.71	18.82	6.99	19.01	6.41	18.54
SepEDA + Diar	6.01	18.89	6.33	18.88	5.70	18.67
SepDiarDA	**7.07**	**16.22**	**7.06**	**16.50**	**6.56**	**15.70**

TABLE III
RESULTS ON COMBINED LIBRI2MIX AND LIBRI3MIX

# spkrs	System	SI-SDRi	DER	SCA
Est	EEND-SS [7]	8.87	6.04	98.2
	EEND-EDA [7]	–	10.16	86.2
	SepEDA	13.05	–	95.3
	SepDiarDA	**13.39**	**4.92**	96.2
Oracle	SepFormer	13.47	–	–
	SepEDA	13.52	–	–
	SepDiarDA	**13.57**	**2.80**	–

results to SepFormer, with a slight degradation observed only in the no-overlap condition. We further examine the impact of integrating mask-based diarization estimation, an approach applicable to most separation models. In the seventh and eighth rows, we present SepFormerS and SepEDA extended with diarization estimation (denoted "+ Diar"). This extension does not affect separation performance, while significantly improves diarization accuracy compared to other baseline architectures. Compared to EEND-SS, the proposed models achieve better performance for all test sets and achieve top performance in both separation and diarization. The last column presents the diarization results for the CH set. The models were initialized with SparseLibri2Mix, adapted with the simulated mixtures from CH and fine-tuned only for diarization with the 2-speaker train part of CH. The proposed systems not only perform best, but also similar to the state-of-the-art diarization models (e.g. DER = 8.35% for EEND-EDA in [4]). EEND-EDA (Ours) performs worse than that due to a much smaller size of the training set: while the standard diarization dataset has ∼2500 h [14], SparseLibri2Mix has ∼60 h. Moreover, unlike standard EEND diarization systems, the training scheme is designed for the joint task. Note that, similar to other separation systems, the proposed methods are limited to processing only relatively short recordings compared to standard diarization systems.

B. LibriheavyMix for Fixed-Speaker Condition

Table II shows the performance on the LibriheavyMix and its test sets: dev, test-clean, and test-other which contain 2–4 speakers. For each speaker condition, the models were trained with the corresponding speaker number. The reported results present the average over the results for the varying speaker number. To keep the comparison between models fair, we selected SepFormerS and SepEDA with diarization ("+ Diar"). The first row presents the results from the dataset paper [19] for a pretrained pyannote system. As pyannote was not fine-tuned on LibriheavyMix, its performance was significantly worse than that of the other models. Among the models we trained, SepDiarDA achieves the best results in both metrics. Interestingly, unlike in previous experiments, SepEDA performs similarly to or slightly worse than SepFormerS. Although SepDiarCA supports a flexible number of speakers, we exclude its results from Tables II and III. In recordings with high or full speech overlap, the absence of single-speaker segments hinders the model's ability to form reliable speaker clusters from filtered embeddings, limiting its effectiveness.

C. Libri2Mix and Libri3Mix for Flexible-Speaker Condition

Table III shows the results for a flexible, both estimated (Est) and Oracle (i.e. known), speaker number. For all methods, the corresponding model trained on Libri2Mix was used for initialization. As training data, we combine Libri2Mix and Libri3Mix train sets. In this final experiment, we present models with their original size and task. Since SepFormer does not support speaker counting, its results are only for the Oracle. Both SepEDA and SepDiarDA provided strong and mutually comparable results. SepDiarDA achieves a significantly better DER value compared to the EEND-SS. The last three rows show the results for the Oracle number of speakers, where SepDiarDA performs better than SepEDA and SepFormer.

VI. CONCLUSION

This letter proposes two models, SepDiarCA and SepDiarDA, which incorporate non-autoregressive attractor generation mechanisms for joint speaker diarization and separation. SepDiarCA leverages speaker information from single-speaker regions, while SepDiarDA enables attractor estimation in highly overlapping speech. Experimental results demonstrate their effectiveness in both conversational and high-overlap scenarios, outperforming existing end-to-end neural models for the joint task and matching or exceeding the performance of models designed for the individual tasks.

ACKNOWLEDGMENT

The authors gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Centers: ACK Cyfronet AGH) for providing computer facilities and support within computational Grants PLG/2023/016823 and PLG/2024/017659.

References

[1] X. Fang et al., “A deep analysis of speech separation guided diarization under realistic conditions,” in *Proc. Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, Tokyo, Japan, 2021, pp. 667–671.

[2] D. Raj et al., “Integration of speech separation, diarization, and recognition for multi-speaker meetings: System description, comparison, and analysis,” in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Shenzhen, China, 2021, pp. 897–904.

[3] H. Taherian and D. Wang, “Multi-channel conversational speaker separation via neural diarization,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 32, pp. 2467–2476, 2024.

[4] S. Horiguchi, Y. Fujita, S. Watanabe, Y. Xue, and K. Nagamatsu, “End-to-end speaker diarization for an unknown number of speakers with encoder-decoder based attractors,” in *Proc. Interspeech* 2020, pp. 269–273.

[5] S. R. Chetupalli and E. A. P. Habets, “Speaker counting and separation from single-channel noisy mixtures,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 31, pp. 1681–1692, 2023.

[6] C. Subakan, M. Ravanelli, S. Cornell, M. Bronzi, and J. Zhong, “Attention is all you need in speech separation,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Proces.*, Toronto, ON, Canada, 2021, pp. 21–25.

[7] S. Maiti et al., “EEND-SS: Joint end-to-end neural speaker diarization and speech separation for flexible number of speakers,” in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2023, pp. 480–487.

[8] Y. Luo and N. Mesgarani, “Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 8, pp. 1256–1266, Aug. 2019.

[9] C. Boeddeker, A. S. Subramanian, G. Wichern, R. Haeb-Umbach, and J. L. Roux, “TS-SEP: Joint diarization and separation conditioned on estimated speaker embeddings,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 32, pp. 1185–1197, 2024.

[10] I. Medennikov et al., “Target-speaker voice activity detection: A novel approach for multi-speaker diarization in a dinner party scenario,” in *Proc. Interspeech*, 2020, pp. 274–278.

[11] J. Kalda, C. Pagés, R. Marxer, T. Alumäe, and H. Bredin, “PixIT: Joint training of speaker diarization and speech separation from real-world multi-speaker recordings,” in *Proc. Speaker Lang. Recognit. Workshop*, Quebec City, QC, Canada, Jun. 2024, pp. 115–122.

[12] Y. Luo, Z. Chen, and T. Yoshioka, “Dual-path RNN: Efficient long sequence modeling for time-domain single-channel speech separation,” in *Proc. ICASSP 2020-2020 IEEE Int. Conf. Acoust. Speech Signal Proces.*, Barcelona, Spain, 2020, pp. 46–50.

[13] M. Rybicka, J. Villalba, N. Dehak, and K. Kowalczyk, “End-to-end neural speaker diarization with an iterative refinement of non-autoregressive attention-based attractors,” in *Proc. Interspeech*, 2022, pp. 5090–5094.

[14] M. Rybicka, J. Villalba, T. Thebaud, N. Dehak, and K. Kowalczyk, “End-to-end neural speaker diarization with non-autoregressive attractors,” *Proc. IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 32, pp. 3960–3973, 2024.

[15] K. Kinoshita, M. Delcroix, and N. Tawara, “Integrating end-to-end neural and clustering-based diarization: Getting the best of both worlds,” in *ICASSP 2021-2021 IEEE Int. Conf. Acoust. Speech Signal Process.*, Toronto, ON, Canada, 2021, pp. 7198–7202.

[16] K. Kinoshita, M. Delcroix, and N. Tawara, “Advances in integration of end-to-end neural and clustering-based diarization for real conversational speech,” in *Proc. Interspeech 2021*, 2021, pp. 3565–3569.

[17] J. Cosentino, M. Pariente, S. Cornell, A. Deleforge, and E. Vincent, “LibriMix: An open-source dataset for generalizable speech separation,” 2020, *arXiv:2005.11262*.

[18] M. Przybocki and M. Alvin, “NIST Speaker Recognition Evaluation LDC2001S97. Philadelphia: Linguistic Data Consortium,” 2000. [Online]. Available: https://catalog.ldc.upenn.edu/LDC2001S97

[19] Z. Jin et al., “LibriheavyMix: A 20,000-Hour dataset for single-channel reverberant multi-talker speech separation, ASR and speaker diarization,” in *Proc. Interspeech 2024*, 2024, pp. 702–706.

[20] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: An ASR corpus based on public domain audio books,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, South Brisbane, QLD, Australia, 2015, pp. 5206–5210.

[21] G. Wichern et al., “WHAM!: Extending speech separation to noisy environments,” in *Proc. Interspeech*, 2019, pp. 1368–1372.

[22] J. L. Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, “SDR–half-baked or well done?,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Proces.*, Brighton, U.K., 2019, pp. 626–630.