IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Traitement du Signal
Vol. 42, No. 5, October, 2025, pp. 2745-2754
Journal homepage: http://iieta.org/journals/ts

A Conditional Generative Adversarial Network for Overlap-Aware Speaker Diarization

Meriem Hamouda[1,2*], Halima Bahi[1,2], Hamza Frihia[3]

[1] Complex Systems Engineering Laboratory (LISCO), Badji Mokhtar-Annaba University, Annaba 23000, Algeria
[2] Computer Science Department, Badji Mokhtar - Annaba University, Annaba 23000, Algeria
[3] Computer Science Department, Ferhat Abbas Setif 1 University, Setif 19000, Algeria

Corresponding Author Email: meriem.hamouda@univ-annaba.dz

https://doi.org/10.18280/ts.420525

Received: 22 May 2025
Revised: 22 August 2025
Accepted: 12 September 2025
Available online: 31 October 2025

Keywords:
speech segmentation, speaker diarization, clustering, self-supervised learning, cGAN

ABSTRACT

In audio processing, speech diarization, also known as speaker diarization, is a technique that answers the question 'Who spoke when?' by automatically dividing an audio stream into segments according to the speaker's identity. It essentially labels every section of the recording with the speaker at any given time. When the speakers involved are known, this processing can be done effectively using end-to-end neural diarization (EEND) methods; however, when the speakers are unknown, the process faces a significant challenge, namely speaker overlap. In such situations, the use of unsupervised approaches is required, particularly with advances in deep generative model architectures. This study pertains to unsupervised speaker diarization amidst multi-speaker contexts; it mainly deals with the detection and allocation of overlapping speech segments, handled in an unsupervised manner, using a conditional generative adversarial network (cGAN). The potential of the proposed method was confirmed during its evaluation on the English part of the CallHome dataset, and the results obtained demonstrate a clear advantage over the hierarchical agglomerative clustering (HAC) algorithm with the VBx-HMM re-segmentation reference method.

1. INTRODUCTION

The speaker diarization (SD) method divides a speech flow into temporal frames, each of which corresponds to a distinct speaker. It is involved in a broad spectrum of applications, ranging from audio/video database indexing to security applications. In real-world scenarios, as multiple speakers may talk simultaneously, this seemingly simple task becomes complicated.

The standard SD process consists of many phases [1], mainly pre-processing, segmentation, and post-processing. The first step is signal pre-processing, where the audio signal is enhanced to improve quality, reduce noise, and remove artifacts that could interfere with diarization. Herein, voice activity detection (VAD) is an essential technique, as it identifies the speech-containing areas in the recordings. VAD algorithms analyze the acoustic features of the audio signal to distinguish speech from background noise or silence [2]. Subsequently, specific features are extracted from the speech signal, capturing the speakers' discriminative features. The commonly utilized features include handcrafted ones, such as the Mel frequency cepstral coefficients (MFCCs), and the embedding representations, like d-vectors and x-vectors [1, 3]. In particular, x-vectors have gained popularity in SD as they are extracted using deep neural networks that robustly capture the speaker characteristics [4]. X-vectors provide a high-level representation of speech segments that are particularly effective for clustering and classification, outperforming the traditional handcrafted features in various scenarios.

After the pre-processing step, the speech signal is segmented into short frames to create homogeneous regions for further analysis [1]. Then, each segment is assigned to a specific class, standing for the related speaker identity. When the dataset lacks annotation, clustering algorithms are deployed to gather comparable segments without prior knowledge of the involved speakers. Standard techniques include hierarchical agglomerative clustering (HAC), K-means clustering, and Gaussian mixture models (GMMs) [5].

Finally, post-processing techniques refine the speaker's boundaries by analyzing prosodic features, language patterns, or contextual cues to re-segment the speech. The state-of-the-art re-segmentation algorithm is the variational Bayesian hidden Markov model (VB-HMM), which combines clustering and re-segmentation. However, although clustering methods have shown effectiveness in various scenarios, they still face challenges such as false alarms, missed detections, and overlapping speaker segments [6], as each cluster is expected to draw the profile of a single speaker.

Overlapping speech detection refers to identifying segments within a recording where two or more speakers are speaking simultaneously, causing their speech signals to overlap [7]. Several approaches have been used to detect overlapping speakers. One common method is thresholding, where a threshold is set based on the extracted features to identify segments where energy levels or other characteristics indicate the presence of speech overlaps. Lately, deep neural networks

have been widely used to differentiate between single-speaker and overlapping regions. Meanwhile, the assignment of the speech to the involved speakers in these regions is more challenging.

The present work tackles the allocation of the overlapping speech segments encountered throughout the diarization post-processing stage. This step is preceded by a pre-processing phase using the VAD algorithm on x-vectors features, which enabled us to eliminate background noise and silences. We then fed the obtained x-vectors into a deep autoencoder to separate frames with speech overlap from those without. In the no-overlap case, diarization is performed efficiently using a clustering algorithm. In the other case, a cGAN is used to separate speakers in mixed speech in an unsupervised manner.

A generative adversarial network (GAN) is a deep neural network trained in an unsupervised way to generate new data that resembles real samples. The cGAN is a GAN that generates new samples while considering additional conditioning knowledge. In our case, the additional knowledge is the profile of each involved speaker issued from the HAC-based clustering; thus, the cGAN is expected to extract the contribution of only one target speaker from a mixed speech. To the best of our knowledge, this is the first application of a cGAN specifically dedicated to the unsupervised allocation of overlapped speech segments within a SD framework. The purpose of this study is to evaluate the effectiveness of the cGAN in improving the state-of-the-art VBx-HMM with a clustering-based method for overlap-aware SD.

The remainder of the paper is organized as follows: Section 2 reviews the relevant prior work. Section 3 presents a detailed description of the proposed self-supervised system for speaker overlap-aware diarization. Section 4 reports and discusses the experimental results. Finally, a conclusion is drawn.

2. RELATED WORK

In recent years, significant progress has been made in the development of systems for detecting and allocating overlapping speech for SD by leveraging deep learning approaches. Herein, end-to-end neural diarization (EEND) systems that address the SD as a multi-class classification problem have met great success [8]. The EEND models are frequently combined with speech separation methods that use advanced deep learning models such as the end-to-end time-domain audio separation network (TasNet), leading to more powerful diarization systems [9]. Target speaker voice activity detection (TS-VAD) is another method to address the overlap-aware SD, where a single speaker is tracked over the mixed speech [10]. Nevertheless, both EEND systems, the separation and TS-VAD systems, are trained in a supervised way and require a huge amount of labeled data.

Meanwhile, clustering-based approaches yield good results in SD, but they show limitations in the case of speech overlaps; indeed, conventional methods struggle to accurately attribute segments to individual speakers. Herein, features extracted from overlapping regions can be ambiguous and may fail to represent a single speaker distinctly [3]. Thus, clustering-based methods focus on the use of discriminative features by using embedding representations [5, 6, 11]. Broadly, the re-segmentation stage operates on x-vectors; therefore, the VBx-HMM re-segmentation algorithm becomes the current trend [12]. VBx uses the x-vector with two-stage clustering: in the first stage, the HAC algorithm performs the clustering, leading to a first speaker-based segmentation; in the second stage, the VBx-HMM is used to refine the current boundaries [6]. Although VBx-HMM achieves good results, “it still cannot properly handle overlapping segments” [13]; therefore, many studies have sought to overcome this limitation. Pal et al. [14] leveraged the labeled data to train a generative adversarial network to strengthen the speakers’ embeddings. X-vector embeddings from short audio clips are used as real input for the GAN discriminator. Zhang et al. [15] proposed a diarization method that handles long-time audio with low latency in real-time scenarios and fixes the inconsistency label issue. It utilizes the chkpt-HAC, a variant of the HAC, to cluster the speakers. The post-processing stage utilized a graph-based re-clustering algorithm.

3. SELF-SUPERVISED OVERLAP-AWARE SD SYSTEM

Figure 1 depicts the several steps involved in the proposed SD method that separates each speaker's contribution when there are overlaps. First, embeddings are extracted from the speech segments after dividing the speech flow into temporal frames. Subsequently, an autoencoder identifies and separates the overlapped from the non-overlapped speech regions. Afterward, the embeddings from the non-overlapped areas are grouped into clusters using GMMs and the HAC algorithm. Finally, the vocal overlap areas are separated by a cGAN trained on clustering results, which extracts the speaker's contribution from the input vocal overlap. The cGAN is called upon as many times as there are contributors to the conversation.

3.1 Embedding representation

As the standard in SD tasks, the x-vector is assumed to be the acoustic representation of the input signals in the proposed system. For this work, x-vector embeddings are obtained using a pre-trained time-delay neural network (TDNN) model provided by SpeechBrain (a speech processing toolkit accessible via https://huggingface.co/speechbrain/spkrec-xvect voxceleb) [16]. The TDNN outputs a 512-dimensional embedding that captures the vocal characteristics of each segment, providing a compact and efficient representation of the speaker identity. These embeddings are extracted from segmented audio input using a fixed-length sliding window, allowing local speaker traits to be embedded into a discriminative vector space.

3.2 Autoencoder for overlapping speech detection

The detection of overlapping speech regions involves identifying speech segments where two or more speakers are speaking simultaneously, resulting in their voices overlapping. To address this issue, we propose the use of an autoencoder. An autoencoder is a deep neural network (DNN) designed to learn the hidden representation of input data through a process of reconstruction. It consists of an encoder that compresses the input into a lower-dimensional latent space and a decoder that reconstructs the input from this latent representation. The effectiveness of the autoencoder is measured by the reconstruction error, which quantifies the difference between the original input and its reconstruction. A smaller

reconstruction error indicates a greater similarity between the input and the output, reflecting the model's ability to capture the hidden patterns of the data [17].

In the present work, the implemented autoencoder is based on a fully convolutional architecture, suited for processing x-vector inputs (reshaped to fit 1D layers). It consists of three successive convolutional layers, each followed by batch normalization and downsampling, standing for the encoder. The decoder performs the inverse operation using convolution and upsampling layers. The goal is to reconstruct the non-overlapping representations faithfully. Table 1 presents the architecture of the used autoencoder.

The goal of using the autoencoder is to train it to distinguish frames that do not represent an overlap from those that do. To this end, during the training phase, we provided it with approximately 150,000 segments. Each segment represents an x-vector representation of a speech recording of a single speaker from the CallHome corpus (NIST SRE 2000) corpus. To optimize the effectiveness of our model, the Mean Squared Error (MSE) between the input vectors and their reconstructions was minimized.

A key challenge lies in defining the decision threshold for reconstruction errors. For this purpose, we compared three strategies-fixed threshold, mean plus two standard deviations, and median absolute deviation (MAD) - directly on the distribution of reconstruction errors obtained from the validation partition. Table 2 shows the comparison of different thresholding strategies for overlap detection.

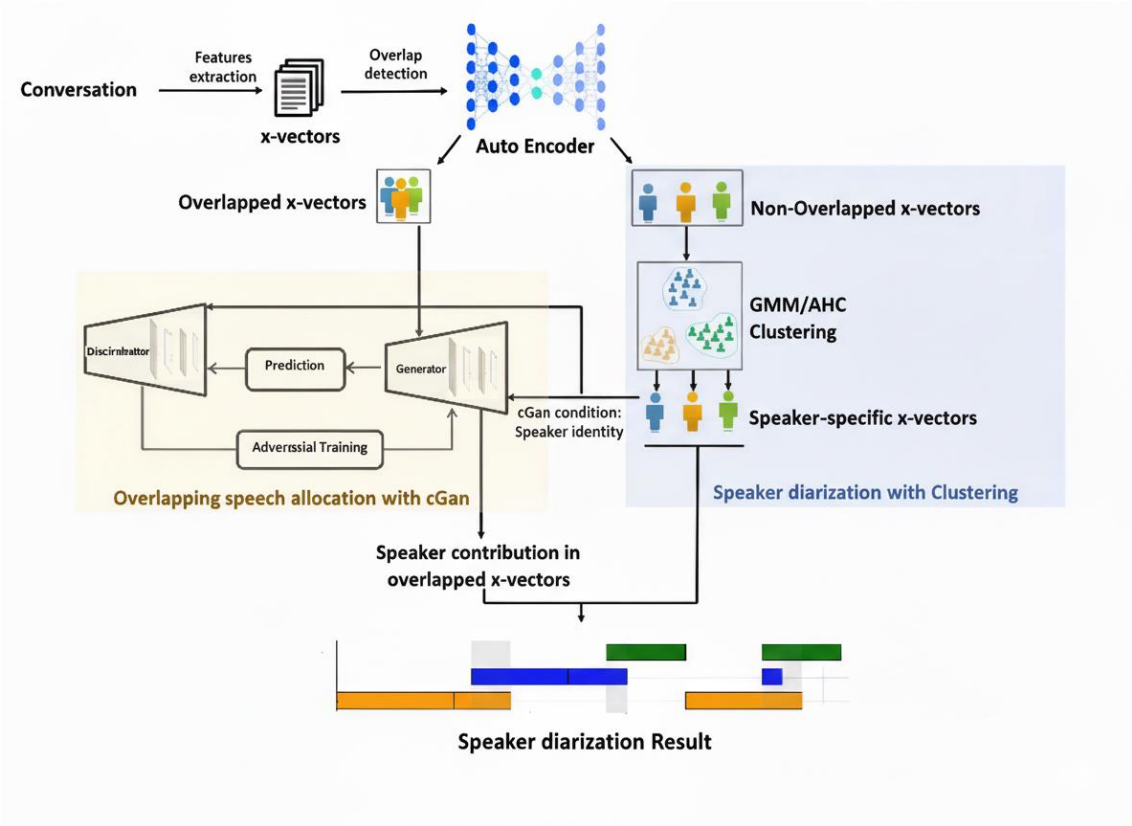


Figure 1. Proposed flow diagram for overlap-aware SD

Table 1. The proposed autoencoder structure

Component	Layer Type	Description	Role
Encoder-Layer1	Conv1D + BatchNorm	32 filters, kernel size 3, ReLU activation, same padding	Captures local patterns in temporal vectors (initial feature extraction)
Encoder-Layer2	1D Max Pooling	Pool size 2	Reduces the temporal dimension (compression)
Encoder-Layer3	Conv1D + BatchNorm	16 filters, kernel size 3, ReLU activation, same padding	Extracts deeper and more abstract patterns
Encoder-Layer4	1D Max Pooling	Pool size 2	Additional compression of information
Bottleneck	Conv1D + BatchNorm	8 filters, kernel size 3, ReLU activation, same padding	Compact latent representation – encodes essential information
Decoder-Layer1	1D Up Sampling	Upsampling factor 2	Reconstructs the temporal dimension
Decoder-Layer2	Conv1D + BatchNorm	16 filters, kernel size 3, ReLU activation, same padding	Progressive reconstruction of details from the latent representation
Decoder-Layer3	1D Up Sampling	Upsampling factor 2	Increases dimension to recover original size
Decoder-Layer4	Conv1D + BatchNorm	32 filters, kernel size 3, ReLU activation, same padding	Refines the reconstruction
Output layer	Conv1D	1 filter, kernel size 3, linear activation, same padding	Final generation of the reconstructed signal (should resemble input if no overlap)

Table 2. Overlap detection & threshold comparison

Thresholding Method	Precision (%)	Recall (%)	F1 (%)	Area Under the Precision – Recall Curve (AUPRC)
Fixed threshold (0.05)	71.2	63.5	67.1	0.64
Mean + 2σ	74.5	66.8	70.4	0.68
MAD (median + 3·MAD)	81.3	74.6	77.8	0.75

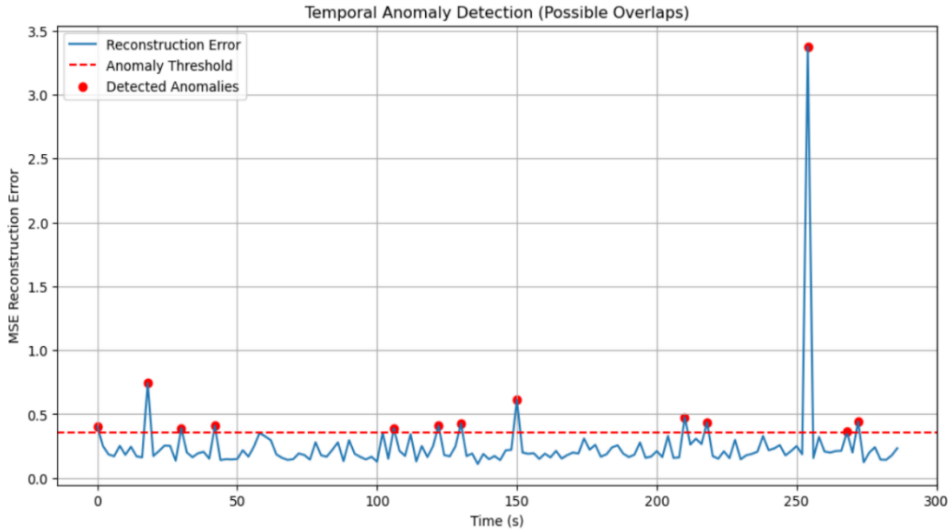


Figure 2. Illustration of overlapping speech detection on the CallHome #4537 recording with a self-supervised autoencoder

Among these, the MAD-based threshold provides the best result, as it adapts to the variability of the error distribution without requiring reference labels. This robustness to outliers and changing acoustic conditions motivated the adoption of MAD in the proposed system.

During inference, the autoencoder is applied to all speech segments extracted from a recording. Segments corresponding to overlapping speech yield reconstruction errors above the adaptive threshold, as illustrated in Figure 2. This confirms the ability of the proposed approach to detect overlap in an unsupervised manner, while leveraging robust thresholding and optimized autoencoder design.

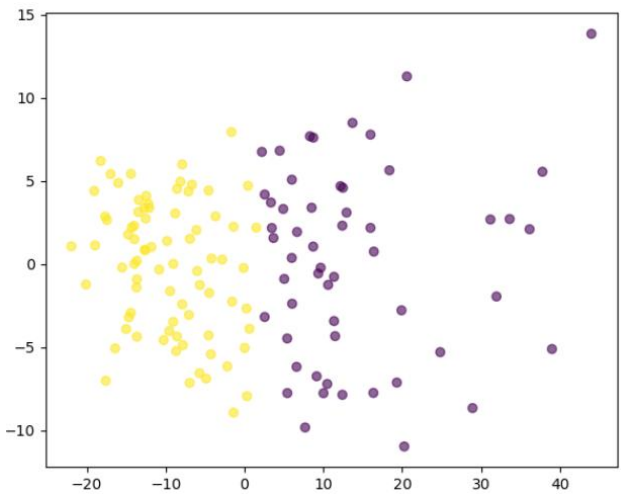
3.3 Clustering

The segments evaluated as non-overlapped, during the overlap detection step, are fed to the clustering algorithm to draw the profile of the involved speakers. The primary objective of clustering algorithms is to group data into distinct clusters, where data points within the same cluster are more similar to each other than those in different clusters. Herein, the HAC algorithm begins with N singleton clusters. The similarity between these clusters is then computed, and each pair of groups with the highest similarity is merged. The process of merging clusters continues until a threshold is reached. A hierarchy of clusters is formed, which serves to identify the number of speakers in a recording and to assign segments to specific speakers [3].

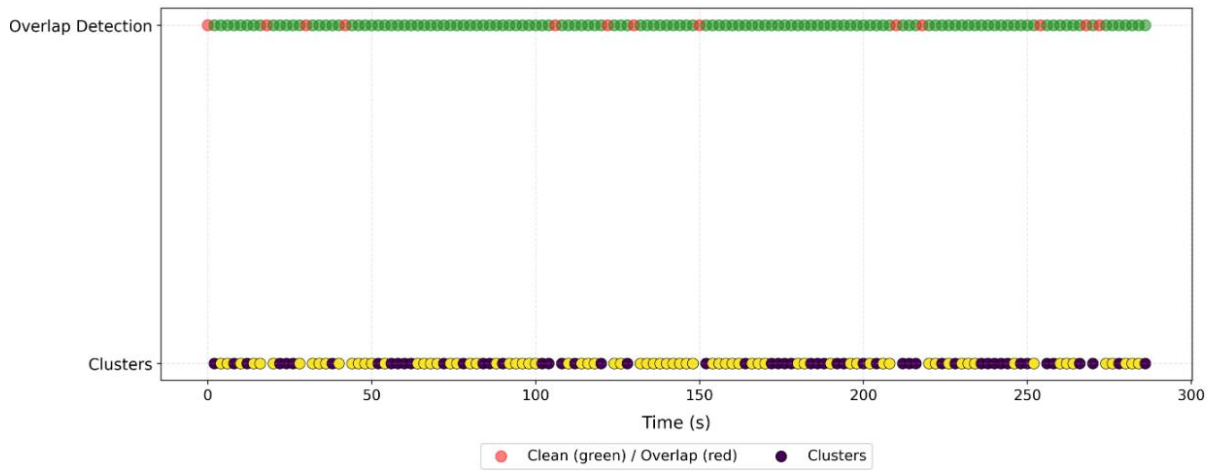
In our suggestion, the HAC algorithm is used in conjunction with the GMM modeling. GMMs model the distribution of acoustic features within each cluster, addressing intra-cluster variance by assuming that data within each cluster follows a mixture of Gaussian distributions. The efficiency of the clustering approach on segments without overlap is seen in Figure 3.

(a) The result of the clustering.

(b) The result of the diarization. The red dots in the top line refer to segments detected as speech overlaps; they were not submitted to the clustering algorithm.



(a)



(b)

Figure 3. Visualization of SD based on the clustering results for CallHome #4537 recording

3.4 A cGAN for overlapping speech allocation

Generative adversarial networks are inspired by a two-player game [18] in which a generator (G) aims to fool a discriminator (D) by producing synthetic samples that resemble real ones, while the discriminator attempts to distinguish real data from generated data. Both components are trained in an adversarial manner. The conditional GAN (cGAN) extends this framework by incorporating additional side information that conditions the generation process, thereby guiding the model towards more plausible outputs.In the context of SD, we propose a cGAN operating directly in the x-vector embedding space to disentangle speakers in overlapping speech segments. The central idea is to learn a mapping such that, for a given x-vector from an overlapping segment (z) and a speaker profile (c), the generator produces an x-vector (ŷ), estimating the contribution of the target speaker to the overlap.

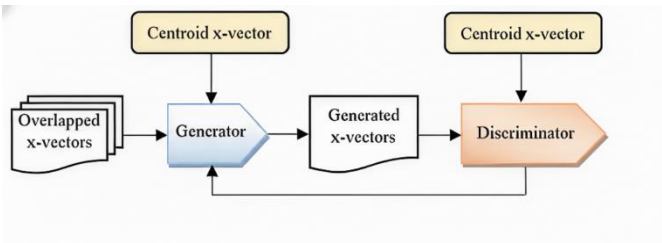


Figure 4. The learning process for generating a single speaker contribution from overlapping speech using a cGAN

The speaker profile (c) is obtained from the clustering stage. All non-overlapping segments assigned to the same cluster are averaged to compute a centroid x-vector, which serves as a robust representation of a speaker's identity. Thus, each cluster acts as a speaker prototype that conditions the generation process. This design ensures that the cGAN leverages unsupervised structure induced by clustering, rather than relying on ground-truth overlap annotations. Figure 4 illustrates the overall mechanism of the generation process.

3.4.1 Inputs, outputs, and conditioning mechanism

The generator takes as input an overlapped x-vector (z, 512 dimensions) alongside with a conditioning vector (c, 512 dimensions), which corresponds to the centroid of a speaker cluster estimated from non-overlapped regions. Outputs a 512-dimensional x-vector G(c,z) that approximates the target speaker's contribution to the overlapped segment.

The discriminator, in turn, receives either a real pair (c,y) - where y denotes a real single-speaker x-vector - or a synthetic pair (c,G(c,z)), and learns to distinguish genuine samples from generated ones, conditioned on the reference profile. This design enforces consistency between the generated embeddings and the target speaker identity. Table 3 summarizes the inputs, outputs, and conditioning mechanism of the proposed cGAN framework.

3.4.2 Network components

The generator is based on a 2D U-Net architecture, which is particularly suited for cGAN frameworks due to its encoder-decoder structure and skip connections that facilitate information flow across layers. Unlike autoencoders, the U-Net does not merely reconstruct its input; Rather, it generates new embeddings conditioned on external knowledge. The discriminator is a convolutional network designed to assess the authenticity of generated embeddings while ensuring their coherence with the conditioning vector. Table 4 reports the architectural components of the proposed cGAN.

3.4.3 Training data and learning strategy

During the training stage, the cGAN is fed with three types of data:

Non-overlapping speech x-vectors: used both as real examples for the discriminator and to compute centroid profiles for conditioning.

Table 3. Inputs, outputs, and conditioning mechanism of the proposed cGAN framework

Component	Input	Output
Generator	Overlapped x-vector z (512) + Conditioning vector c (512)	Generated x-vector G(c,z) (512)
Discriminator (Real pair)	Real x-vector y (512) + Conditioning vector c (512)	Label = Real
Discriminator (Synthetic pair)	Generated x-vector G (c, z) (512) + Conditioning vector c (512)	Label = Fake

Table 4. The cGAN's architecture

Component	Layer Type	Parameters	Output Shape	Activation	Purpose
Generator	Input	Overlap (512) + Condition (512)	(1024)	-	Concatenated input
	Reshape	-	(16, 64, 1)	-	Prepares for 2D convs
Encoder	Conv2D	64 filters, 4 × 4, stride 2	(8, 32, 64)	LeakyReLU	Feature extraction
	Conv2D	128 filters, 4 × 4, stride 2	(4, 16, 128)	LeakyReLU	Downsampling
Bottleneck	Conv2D	256 filters, 4 × 4, stride 2	(2, 8, 256)	LeakyReLU	Latent representation
Decoder	Conv2DTranspose	128 filters, 4 × 4, stride 2	(4, 16, 128)	ReLU, BN	Upsampling
	Conv2DTranspose	64 filters, 4 × 4, stride 2	(8, 32, 64)	ReLU, BN	Further upsampling
	Conv2DTranspose	32 filters, 4 × 4, stride 2	(16, 64, 32)	ReLU	Final upsampling
	Conv2D	1 filter, 3 × 3	(16, 64, 1)	Tanh	Projection
	Flatten + Dense	512 units	(512)	Linear	Final x-vector output
Discriminator	Input	Candidate (512) + Condition (512) → concat	(1024)	-	Input
	Reshape	-	(16, 64, 1)	-	Preparation
	Conv2D	64 filters, 4 × 4, stride 2	(8, 32, 64)	LeakyReLU	Feature extraction
	Conv2D	128 filters, 4 × 4, stride 2	(4, 16, 128)	LeakyReLU	Feature extraction
	Conv2D	256 filters, 4 × 4, stride 2	(2, 8, 256)	LeakyReLU	Final features
	Flatten + Dense	1 unit	(1)	Sigmoid	Real/fake decision

Overlapping speech x-vectors (real data): provided as inputs to the generator, without ground-truth targets.

Synthetic overlaps: created by mixing pairs of non-overlapping x-vectors with random weights. Since the ground-truth components are known, these samples enable the addition of a reconstruction loss that compares generated outputs to the true x-vectors of individual speakers. To generate these synthetic overlaps in practice, we randomly select pairs of non-overlapping x-vectors from different speakers and combine them linearly with a mixing coefficient α drawn from a uniform distribution in [0.3, 0.7], ensuring both speakers contribute to the mixture.

3.4.4 Objective function

Given the inputs x and z, z standing for the overlapping speech and x representing the additional knowledge, and an output y standing for a real sample, $G(x, z)$ is the generated sample starting from both x and z. The discriminator outputs a probability indicating the truthfulness of the generator output. When the output is authentic, $D(x, y)$, the probability outputted by the discriminator is close to 1; elsewhere, it is $D(x, G(x, z))$, the probability is close to 0, denoting a synthetic sample. Thus, the objective of a cGAN is expressed as follows [19]:

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}\left[logD(x,y)\right] + \mathbb{E}_{x,z}\left[\log\left(1 - D\left(x, G(x,z)\right)\right)\right] \quad (1)$$

For that purpose, G tries to minimize this objective against an adversarial D that tries to maximize it, leading to:

$$G^* = arg\ min_G\ max_D\ \mathcal{L}_{cGAN}(G, D) \quad (2)$$

In the case of the cGAN, Isola et al. [19] suggested adding a reconstruction loss to reduce the difference between the ground truth and the generated samples.

$$\mathcal{L}_1(G) = \mathbb{E}_{x,y,z}[\|y - G(x,z)\|_1] \quad (3)$$

Therefore, the final objective becomes:

$$G^* = arg\ min_G\ max_D\ \mathcal{L}_{cGAN}(G, D) + \lambda.\mathcal{L}_1(G) \quad (4)$$

where, λ is a weight that balances between the adversarial loss in Eq. (1) and the reconstruction loss in Eq. (3). The weighting factor λ was treated as a hyperparameter and selected empirically using a grid search on the validation set. The chosen value, λ=1, provided a stable training process and an optimal trade-off between generating realistic embeddings and accurately reconstructing the target speaker's contribution in overlapping segments.

3.4.5 Generation of a single speaker contribution

During inference, the cGAN is provided with a speech segment identified as overlapping, along with a representation of the target speaker obtained from clustering. The model is expected to extract the target speaker's contribution from the overlapped region. The resulting speaker identity depends on the given speaker representation. Figure 5 illustrates the separation of four speakers engaged in the same conversation.

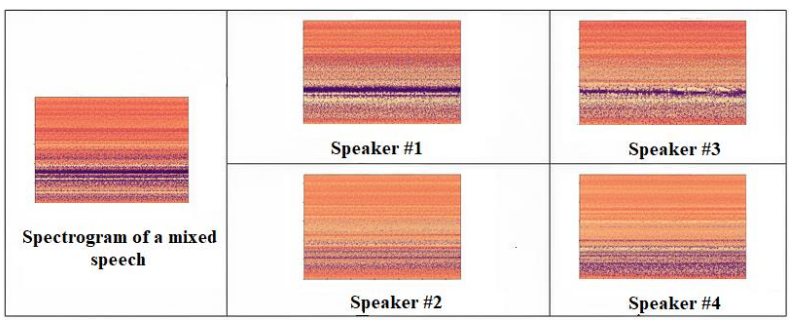


Figure 5. Illustration of the separation from the Callhome #4726 recording

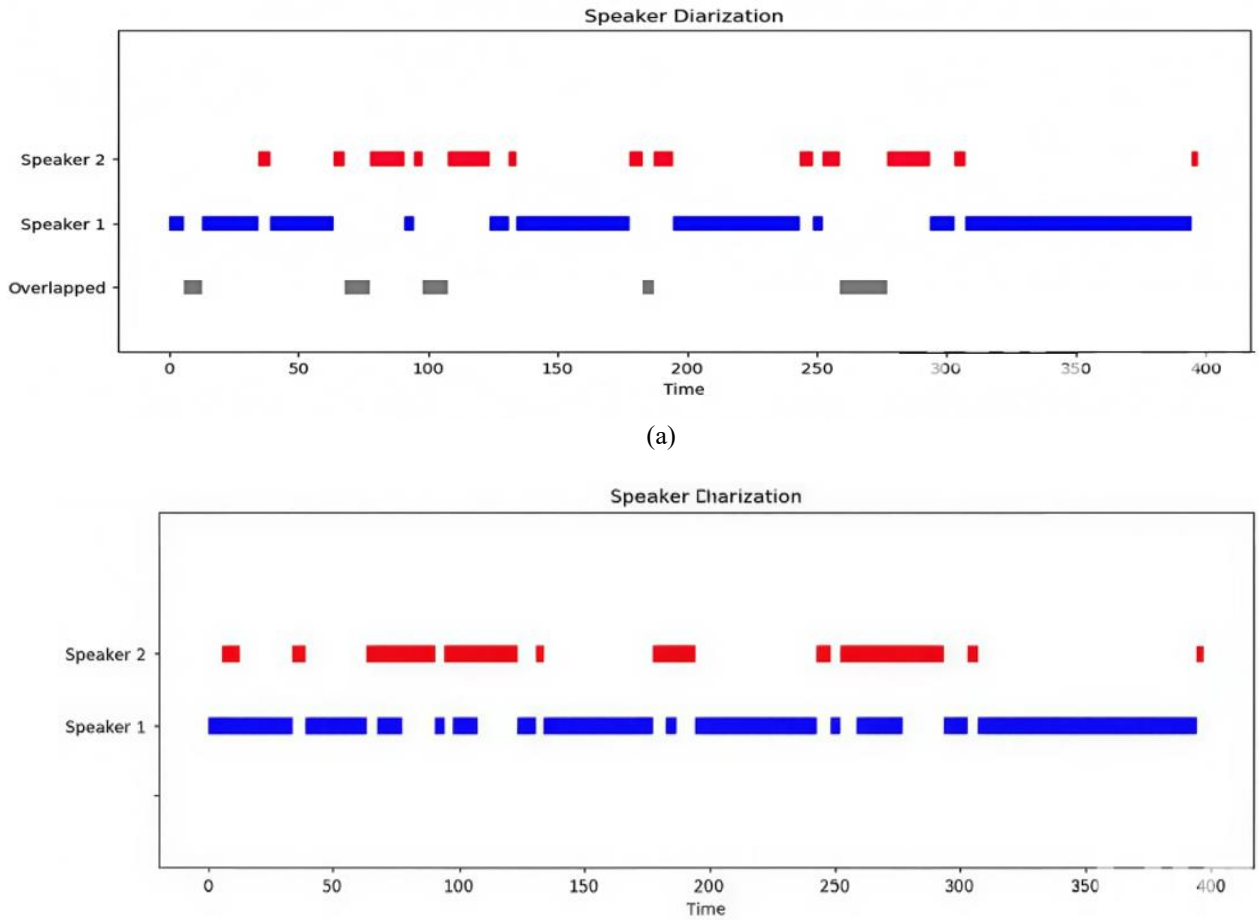


Figure 6. Diarization results as a timesheet

3.5 Diarization and labeling

Once each speaker's contribution has been identified and isolated in the overlapping regions, the labeled speech segments are broken down at the frame level and annotated based on their respective start and end times. Figure 6 illustrates the SD outcome in the form of a timesheet, displaying the overlapped regions before the separation phase and assigning each separated region to its corresponding speaker after the separation phase.

(a) before the overlapped speech assignment

(b) after the overlapped speech assignment

4. RESULTS AND DISCUSSION

To evaluate the effectiveness of the proposed overlap-aware SD system, many experiments were carried out. The first experiments aim to set the generator's suitable architecture for the target task. Once the final architecture of the U-Net model was adopted, the results of the diarization systems were compared with those from SOTA models, particularly, the HAC-based clustering with VBx-HMM re-segmentation. The system's performances are assessed on the CallHome dataset.

4.1 Evaluation dataset

The CallHome English corpus was developed by the Linguistic Data Consortium (LDC); it includes telephone conversations between native English speakers, who represent various demographic categories. Although the participants were aware of the recording, the conversations were unrestricted in terms of topic choice and had no additional limitations. Each telephone exchange lasts approximately 4 to 30 minutes [20].

The corpus contains 176 conversations, of which 164 are two-speaker conversations, ten are three-speaker conversations, and two are four-speaker conversations. The corpus is known to have a high speech overlap rate.

4.2 Results

4.2.1 Comparison of cGAN architectures

The first experiments were conducted to select the best architecture for the U-Net model. Table 5 compares various conditional GAN architectures based on the loss function, varying the generator and discriminator depths.

The architecture, with three layers for both the encoder and decoder components in the U-Net architecture, and with three layers for the discriminator, was retained to pursue our experiments.

4.2.2 Comparison with SOTA diarization systems

Once the architecture of the cGAN was selected, additional experiments were conducted to compare the proposed diarization system results with those obtained with traditional baseline systems in terms of diarization error rate (DER), precision, recall, and F1-score. Table 6 reports the results.

Table 5. Comparison of cGAN architectures based on final training losses

Generator Layers	Discriminator Layers	Loss
1 encoder + 1 decoder	2	1.85
2 encoder + 2 decoder	2	1.42
2 encoder + 2 decoder	3	1.20
3 encoder + 3 decoder	2	0.95
3 encoder + 3 decoder	3	0.78

Table 6. Performance of the proposed method compared to the baseline systems

	DER (%)	Precision (%)	Recall (%)	f1-Score
MFCC + HAC	52.83	56.92	57.43	57.17
x-vector + HAC/GMM	21.04	87.92	88.40	88.16
x-vector + HAC/GMM+VBx-HMM	16.30	90.85	90.53	90.69
Present work	8.56	94.99	94.70	94.84

These results confirm the advantage of combining embedding-based speaker representations with GMM models over the baseline system with HAC and MFCCs. They also demonstrate the improvement achieved by incorporating the re-segmentation stage via the VBx-HMM.

Finally, the results highlight the substantial improvement achieved by our suggestion, which explicitly addresses overlapping speech segments. The integration of the cGAN significantly reduces the DER and improves precision, recall, and F1-score.

4.2.3 Separation quality assessment

The central claim of this work lies in the ability of the cGAN to disentangle speakers in overlapping regions. To evaluate this capability, we measured separation quality using the equal error rate (EER), a common metric in speaker verification that corresponds to the point where false acceptance and false rejection rates are equal. A lower EER reflects better preservation of speaker identity. Table 7 presents the results under overlapped, separated, and non-overlapped conditions, illustrating the impact of the proposed approach.

Table 7. Separation quality evaluation using EER

Condition	EER (%)
Overlapped (without separation)	13.4
After cGAN separation	8.7
Non-overlapped (reference)	5.4

To contextualize these findings, non-overlapped segments serve as a lower bound, showing the best possible performance. As expected, overlapped segments without separation lead to severe degradation, while applying the cGAN substantially reduces the EER, narrowing the gap with the optimal non-overlapped condition.

4.3 Comparison with recent overlap-aware SD approaches

In recent years, several end-to-end diarization methods that explicitly address overlap have emerged, showing competitive performance. Among them, EEND with encoder-decoder attractors (EEND-EDA) [21] and its extensions [22] achieved promising results on meeting and conversational datasets.

However, these approaches are supervised and require training on large annotated corpora, which limits their transferability to contexts or languages with limited labeled data. In contrast, the proposed method does not rely on end-to-end supervised learning; it naturally integrates into a standard diarization pipeline based on x-vectors and clustering, and is applied only to overlapped segments. This selectivity reduces computational cost and avoids the need to retrain a complete model on task-specific data.

The cGAN must indeed be executed for each overlapped segment and for each target speaker, which introduces an additional cost. However, this impact remains limited for two main reasons: first, overlapped segments represent a small fraction of the total signal; second, cGAN training is performed offline, meaning that in practice, only the inference cost needs to be considered. This cost remains compatible with real-world applications.

4.4 Robustness to clustering errors

A key limitation of the proposed method lies in its dependence on the quality of clustering, as the cGAN relies on cluster centroids as conditioning profiles. Two main types of clustering errors can occur.

The over-clustering (a single speaker split into multiple clusters): This case provides the cGAN with several conditioning vectors that all represent the same speaker. While this may reduce efficiency, it is unlikely to critically harm separation, as the generated embeddings remain acoustically consistent with the true identity.

The under-clustering (different speakers merged into one cluster): This represents a more critical failure. The centroid becomes a blend of multiple speakers, and the cGAN may generate embeddings that do not match any true speaker, propagating errors to the final diarization.

A preliminary simulation of such errors confirmed this sensitivity: artificial under-clustering increased DER by approximately 8-10% compared to the ideal case, while over-clustering had a much smaller impact. This indicates that our method is best suited to refine a reasonably good initial clustering rather than to recover from severe clustering errors.

Future work will explore strategies to improve robustness, such as iterative refinement of clustering using cGAN outputs (EM-like re-estimation), or uncertainty-aware conditioning that accounts for cluster compactness.

5. CONCLUSION

This study addressed the challenging task of SD in the presence of overlapping speech, a problem that has gained increasing attention with the growing need for robust conversational analysis. Recent advances, such as EEND and

TS-VAD, have achieved competitive DERs by explicitly modeling overlaps. Yet, these approaches heavily rely on large amounts of annotated data and remain supervised in nature.

In contrast, we explored a cGAN-based approach that reconstructs the contribution of each speaker within overlapping segments, while relying on unsupervised clustering for non-overlapping regions. This makes the method particularly well-suited for scenarios with limited annotated resources.

Our experiments on the CallHome dataset evaluated multiple cGAN architectures, showing that the proposed system significantly improves separation quality compared to baseline methods. Importantly, the separation evaluation using EER demonstrated that our system produces overlaps closer in quality to non-overlapped signals, confirming its robustness. Moreover, when compared with recent overlap-aware approaches, the proposed system achieved competitive results while operating under less restrictive data requirements.

Overall, the proposed framework demonstrates a promising balance between robustness, adaptability to unlabeled settings, and practical applicability, paving the way for overlap-aware diarization systems that do not rely on extensive annotated corpora.

In parallel, we note that the proposed method depends on clustering quality. While it shows resilience to over-clustering, under-clustering remains a more challenging failure case, motivating future work on robustness-aware conditioning strategies.

Finally, although the proposed cGAN-based overlap-aware SD does not reach the ideal performance observed on non-overlapped signals, it provides an interesting compromise: It significantly reduces the gap compared to raw overlapped segments, while remaining applicable in practical scenarios. Future work will explore a more systematic evaluation of computational efficiency as well as hybrid strategies combining the strengths of cGAN-based separation with end-to-end diarization frameworks.

REFERENCES

[1] O'Shaughnessy, D. (2025). Speaker diarization: A review of objectives and methods. Applied Sciences, 15(4): 2002. https://doi.org/10.3390/app15042002

[2] Vesperini, F., Vecchiotti, P., Principi, E., Squartini, S., et al. (2016). Deep neural networks for multi-room voice activity detection: Advancements and comparative evaluation. In 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, Canada, pp. 3391-3398. https://doi.org/10.1109/IJCNN.2016.7727633

[3] Park, T.J., Kanda, N., Dimitriadis, D., Han, K.J., et al. (2022). A review of speaker diarization: Recent advances with deep learning. Computer Speech & Language, 72: 101317. https://doi.org/10.1016/j.csl.2021.101317

[4] Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., et al. (2018). X-vectors: Robust DNN embeddings for speaker recognition. In 2018 IEEE international conference on acoustics, Speech and Signal Processing (ICASSP) Calgary, Canada, pp. 5329-5333. https://doi.org/10.1109/ICASSP.2018.8461375

[5] Hamouda, M., Bahi, H. (2023). Feature embedding representation for unsupervised speaker diarization in telephone calls. Intelligent Systems and Pattern Recognition, 1940: 207-215. https://doi.org/10.1007/978-3-031-46335-8_16

[6] Bullock, L., Bredin, H., Garcia-Perera, L.P. (2020). Overlap-aware diarization: Resegmentation using neural end-to-end overlapped speech detection. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, pp. 7114-7118. https://doi.org/10.1109/ICASSP40776.2020.9053096

[7] Yousefi, M., Hansen, J.H.L. (2020). Frame-Based overlapping speech detection using convolutional neural networks. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, pp. 6744-6748. https://doi.org/10.1109/ICASSP40776.2020.9053108

[8] Fujita, Y., Watanabe, S., Horiguchi, S., Xue, Y., et al. (2020). End-to-end neural diarization: Reformulating speaker diarization as simple multi-label classification. arXiv preprint, arXiv:2003.02966. https://doi.org/10.48550/arXiv.2003.02966

[9] Fang, X., Ling, Z.H., Sun, L., Niu, S.T., et al. (2021). A deep analysis of speech separation guided diarization under realistic conditions. In Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Tokyo, Japan, pp. 667-671. https://doi.org/10.1109/APSIPAASC52472.2021.9689773

[10] Medennikov, I., Korenevsky, M., Prisyach, T., Khokhlov, Y., et al. (2020). Target-speaker voice activity detection: A novel approach for multi-speaker diarization in a dinner party scenario. In Interspeech 2020, Shanghai, China, pp. 2747-2751. https://doi.org/10.21437/Interspeech.2020-1602

[11] Raj, D., Huang, Z., Khudanpur, S. (2021). Multi-class spectral clustering with overlaps for speaker diarization. In IEEE Spoken Language Technology Workshop (SLT): Shenzhen, China, pp. 582-589. https://doi.org/10.1109/SLT48900.2021.9383602

[12] Diez, M., Burget, L., Wang, S., Rohdin, J., et al. (2019). Bayesian HMM based x-vector clustering for speaker diarization. In Interspeech 2019, Graz, Austria, pp. 346-350. https://doi.org/10.21437/Interspeech.2019-2399

[13] He, M., Raj, D., Huang, Z., Du, J., et al. (2021). Target-speaker voice activity detection with improved i-vector estimation for unknown number of speaker. arXiv preprint arXiv:2108.03342. https://doi.org/10.48550/arXiv.2108.03342

[14] Pal, M., Kumar, M., Peri, R., Park, T.J., et al. (2020). Speaker diarization using latent space clustering in generative adversarial network. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, pp. 6504-6508. https://doi.org/10.1109/ICASSP40776.2020.9053952

[15] Zhang, Y., Lin, Q., Wang, W., Yang, L., et al. (2021). Low-latency online speaker diarization with graph-based label generation. arXiv preprint arXiv:2111.13803. https://doi.org/10.48550/arXiv.2111.13803

[16] Ravanelli, M., Parcollet, T., Plantinga, P., Rouhe, A., et al. (2021). SpeechBrain: A general-purpose speech toolkit. arXiv preprint arXiv:2106.04624. https://doi.org/10.48550/arXiv.2106.04624

[17] Dendani, B., Bahi, H., Sari, T. (2021). Self-supervised speech enhancement for Arabic speech recognition in

real-world environments. Traitement du Signal, 38(2): 349-358. https://doi.org/10.18280/ts.380212

[18] Goodfellow, I., Pouget-Abadie, J., Mirza, B., Xu, B., et al. (2020). Generative adversarial networks. Communications of the ACM, 63(11): 139-144. https://doi.org/10.1145/3422622

[19] Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A. (2017). Image-to-image translation with conditional adversarial networks. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, USA, pp. 5967-5976. https://doi.org/10.1109/CVPR.2017.632

[20] Katerenchuk, D., Brizan, D.G., Rosenberg, A. (2018). Interpersonal relationship labels for the CallHome corpus. In proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018): Miyazaki, Japan, pp. 3749-3723. https://aclanthology.org/L18-1592/.

[21] Fujita, Y., Kanda, N., Horiguchi, S., Xue, Y., Nagamatsu, K., Watanabe, S. (2019). End-to-end neural speaker diarization with self-attention. In 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Singapore, pp. 296-303. https://doi.org/ 10.1109/ASRU46091.2019.9003959

[22] Horiguchi, S., Fujita, Y., Watanabe, S., Xue, Y., et al. (2020). End-to-end speaker diarization for an unknown number of speakers with encoder-decoder based attractors. arXiv preprint arXiv:2005.09921. https://doi.org/10.48550/arXiv.2005.09921