Continual Self-supervised Domain Adaptation for End-to-end Speaker Diarization

Juan M. Coria[1], *Hervé Bredin*[2], *Sahar Ghannay*[1], *Sophie Rosset*[1]

[1]Université Paris-Saclay CNRS, LISN, Orsay, France
[2]IRIT, Université de Toulouse, CNRS, Toulouse, France

{juan-manuel.coria, sahar.ghannay, sophie.rosset}@lisn.upsaclay.fr
herve.bredin@irit.fr

Abstract

In conventional domain adaptation for speaker diarization, a large collection of annotated conversations from the target domain is required. In this work, we propose a novel continual training scheme for domain adaptation of an end-to-end speaker diarization system, which processes one conversation at a time and benefits from full self-supervision thanks to pseudo-labels. The qualities of our method allow for autonomous adaptation (*e.g.* of a voice assistant to a new household), while also avoiding permanent storage of possibly sensitive user conversations. We experiment extensively on the 11 domains of the DIHARD III corpus and show the effectiveness of our approach with respect to a pre-trained baseline, achieving a relative 17% performance improvement. We also find that data augmentation and a well-defined target domain are key factors to avoid divergence and to benefit from transfer.

Index Terms: self-supervised learning, end-to-end speaker diarization, continual learning, domain adaptation

1. Introduction

Speaker diarization aims at determining "who spoke when" in a recorded conversation, partitioning the audio sequence according to speaker identity. Recent *end-to-end* speaker diarization systems [1, 2] have simplified this by training a single neural network in a permutation-invariant manner to ingest an audio recording and produce an *overlap-aware* speaker diarization output. These systems are usually trained to perform well on a given corpus with its own set of specific properties (*e.g.* microphone quality, noise, speaker accent, language, etc.) shared among recordings. We refer to this set of shared properties as a *domain*. However, it is well known that the performance of the same system on a different domain is substantially worse than on the training domain, a problem known as *domain mismatch*. In domain adaptation, the goal is to fix this mismatch by fine-tuning the *out-of-domain* system on the target domain in which we want to obtain good performance. In particular, end-to-end training makes speaker diarization systems suitable for domain adaptation because fine-tuning a single model is simpler than doing so for multiple modules. Nevertheless, domain adaptation remains expensive for two reasons: 1) a relatively large number of new domain conversations needs to be collected, and 2) they need to be manually annotated.

Pseudo-labels [3, 4] were originally designed for semi-supervision (mixing labeled and unlabeled data) by using the predictions of a pre-trained system as annotations in a teacher-student training scheme. However, they are also an interesting alternative to remove the need for annotated data. This method has shown great promise in end-to-end speaker diarization [5], where authors experiment with an iterative and committee-based training scheme. Another similar study on domain adaptation for speech enhancement [6] even goes one step further, showing that periodically updating the teacher model while fine-tuning the student on the target domain can significantly increase performance.

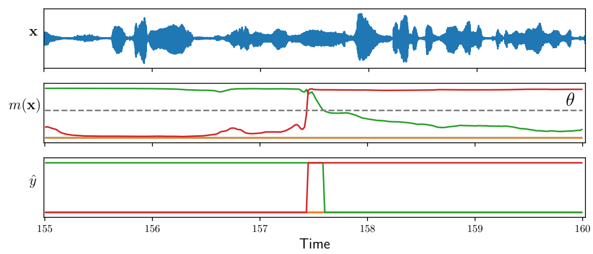

Figure 1: *Real example of a system input* $\mathbf{x}$ *and output* $m(\mathbf{x})$*. Pseudo-labels* $\hat{y}$ *are obtained by binarizing* $m(\mathbf{x})$ *with a fixed threshold* $\theta = 0.5$.

On the other hand, it is possible to eliminate the need to "a priori" collect a large new domain corpus by relying on *continual learning* [7]. This paradigm is defined by a sequential training scheme where new data (individual conversations in our case) become available as time passes. After sequential training on new conversations from a target domain, we expect the system to perform well on both past and future conversations of that domain. As defined in previous work [7, 8], improvement on past conversations is usually referred to as *backward transfer*, while *forward transfer* is used to denote improvement on future conversations. Unfortunately, continual learning is prone to *catastrophic forgetting* [9], whereby performance on past conversations sharply drops as the model is trained on new ones (*i.e.* negative backward transfer). A naive solution is to keep all past conversations for future training. However, storing these conversations permanently may be problematic or even impossible in some cases, as they are usually regarded as sensitive or personal identifiable data.

In this work, we study continual domain adaptation for end-to-end speaker diarization. We propose a fully self-supervised training scheme that achieves an average 17% relative improvement over a pre-trained baseline without a single manually annotated conversation. Our approach also rivals (and sometimes outperforms) non-continual variants trained on the whole target domain at once. Furthermore, since only a single conversation at a time is used for training, every new conversation can be discarded as soon as it is processed, avoiding any potential unwanted access.

2. End-to-end speaker diarization

Following [1, 10], end-to-end speaker diarization is modeled as a multi-label classification problem. In our case, a model m is trained to ingest a 5s audio chunk $\mathbf{x}$ and produce speaker activity probabilities $m(\mathbf{x}) = \{\mathbf{s}_1, \ldots, \mathbf{s}_F\}$ as depicted in Figure 1, where F is the number of output frames and $\mathbf{s}_f \in [0,1]^{K_{\max}}$, with $K_{\max}$ the estimated maximum number of different speakers in an input (in our case $K_{\max} = 4$). Since any permutation of speakers in the output is equivalent in terms of diarization performance, a permutation-invariant loss [1] is minimized:

$$\mathcal{L}(\mathbf{y}, m(\mathbf{x})) = \min_{\text{perm} \in \mathcal{P}} \mathcal{L}_{\text{BCE}}(\text{perm}(\mathbf{y}), m(\mathbf{x})) \tag{1}$$

where $\mathbf{y}$ is the reference annotation (manual or pseudo-label) for $\mathbf{x}$, $\mathcal{L}_{\text{BCE}}$ is the frame-wise binary cross entropy loss and $\mathcal{P}$ the set of all possible speaker permutations of $\mathbf{y}$.

3. Proposed training scheme

In this section, we present the different components of our training scheme as depicted in Figure 2. Specifically, given a model pre-trained on an out-of-domain corpus, we train on one conversation of the target domain at a time using pseudo-labels. We first define self-supervision with pseudo-labels in Section 3.1 and then discuss continual training in Section 3.2.

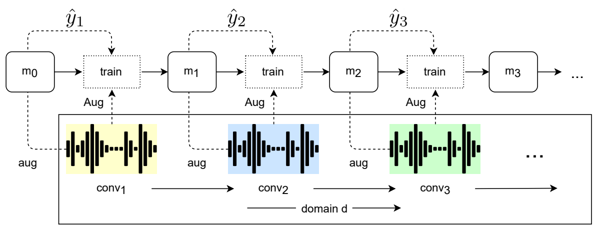

Figure 2: *Continual training over conversations* conv_t *of domain* d *using pseudo-labels* $\hat{y}_t$. *Model* m_0 *pre-trained on an out-of-domain corpus produces the first pseudo-labels* $\hat{y}_1$. *From then onwards, each model* m_{t-1} *produces pseudo-labels* $\hat{y}_t$ *and is then trained only on conversation* conv_t, *resulting in a new model* m_t.

3.1. Self-supervision

Many successful works on self-supervision [11, 12, 13] rely on auxiliary tasks that attempt to predict artificially missing or distorted parts of the input. In contrast, in pseudo-labeling [3] a pre-trained *teacher* model generates labels to train the *student* model. A similar idea has been applied to speaker diarization in [5] using a committee-based method where pseudo-labels are a combination of predictions from multiple systems. Our work is more similar to [6], as the trained model is both teacher and student. However, contrary to [6], we train on a single conversation at a time instead of a large target domain corpus.

3.1.1. Pseudo-labels

Pseudo-labels $\hat{y}$ can be an exact copy of $m(\mathbf{x}) \in [0,1]^{K_{\max}F}$. However, as shown in Figure 1, $m(\mathbf{x})$ can be noisy and could fail to provide a useful training signal in some input regions, so we obtain $\hat{y}$ by binarizing $m(\mathbf{x})$ with a threshold $\theta = 0.5$. As depicted in Figure 2, model m_{t-1} generates pseudo-labels

Subset	Domain	Recordings		Duration		Spk / Rec.		Base
		dev	test	dev	test	dev	test	DER
DH$_A$	Broadcast Interview	12	12	2.1h	2.0h	3.8	3.7	4.6
	Court	12	12	2.1h	2.0h	6.9	7.3	8.9
	Socio Lab	16	12	2.7h	2.0h	2.0	2.0	12.9
	CTS	61	61	10.2h	10.2h	2.0	2.0	20.4
	Meeting	14	11	2.4h	1.9h	5.4	3.9	33.5
	Restaurant	12	12	2.0h	2.1h	7.2	6.4	49.5
DH$_B$	Audiobooks	12	12	2.0h	2.0h	1.0	1.0	5.2
	Maptask	23	19	2.5h	2.1h	2.0	2.0	11.0
	Socio Field	12	22	2.0h	2.3h	3.5	2.3	18.2
	Clinical	48	51	4.3h	4.4h	2.0	2.0	21.6
	Webvideo	30	35	1.9h	2.1h	4.0	4.1	41.8

Table 1: *Information on DIHARD III domains. The average number of speakers per recording ("Spk / Rec.") and the DER of the VB-HMM baseline for track 2 [15] ("Base DER") are evidence of domain differences in difficulty.*

$\hat{y}$ from conversation conv_t that are used to train m_{t-1}, resulting in model m_t. A risk of the model being both teacher and student is divergence, as matching its own predictions may progressively reinforce errors. To limit this, we rely on data augmentation. During training we first calculate pseudo-labels $\hat{y}$ with a weak noise augmentation aug:

$$\hat{y}_{kf} = \begin{cases} 1 & \text{if } \; m_{t-1}(aug(\mathbf{x}))_{kf} \geq \theta \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where k indexes speakers and f indexes frames. Our hypothesis is that generating $\hat{y}$ from a weak perturbation of $\mathbf{x}$ may help to prevent divergence by providing multiple views of the same input, acting as a form of regularization. As depicted in Figure 2, we use $Aug(\mathbf{x}))$ as inputs during training to improve robustness, where Aug is a strong augmentation adding both noise and reverberation.

3.1.2. Stopping criterion

We use the area under the receiver operating characteristic curve (AUROC) [14] as a validation metric when training on each conversation, allowing to measure model improvement. The ROC curve is calculated by relating false positives and true positives at different decision thresholds. Taking the example of Figure 1, this is equivalent to applying multiple thresholds $\theta \in [0,1]$ to $m(\mathbf{x})$ and comparing the result to the pseudo-labels. Notice that this metric may not approximate actual performance correctly if pseudo-labels contain errors.

Since the model has no access to a validation set, we extract 30% of conv_t to form dev_t, which is spread over the whole duration of the recording. We then calculate the AUROC on dev_t after each epoch on the remaining set train_t. When the validation AUROC does not improve for a certain number of epochs, we stop training and wait for the next conversation.

3.2. Continual training

As mentioned previously, continual training is prone to catastrophic forgetting [7, 9]. A naive solution is to train on all $\text{conv}_{\leq t}$ at a given step t, but this introduces two problems. First, the cost of training on conv_t grows linearly with the number of conversations, that can in theory be infinite. Second, storing conversations permanently may not be possible, as it is usually considered sensitive data that needs to be guarded from external access. Other works in continual learning [16, 17] show that it is possible to limit forgetting by keeping a memory buffer with previous inputs or latent features, partially solving the first problem but ignoring the second. Other popular methods [18, 19]

System	Labels?	Continual?	$\mathrm{Aug}(x)$?	**Average**	**Broadcast Interview**	**Court**	**Socio Lab**	**CTS**	**Meeting**	**Restaurant**	**Audiobooks**	**Maptask**	**Socio Field**	**Webvideo**	**Clinical**
pre-trained	NA	NA	NA	51.4	35.9	22.0	64.8	28.3	86.8	47.5	28.2	40.9	56.8	73.0	80.9
ours1	pseudo	✓		56.7	32.2	**21.1**	74.2	25.9	92.3	47.2	27.5	50.1	55.0	99.9	97.9
ours2	pseudo	✓	✓	**42.8**	**30.4**	21.5	**39.3**	**25.3**	**46.8**	44.3	**25.9**	**34.3**	**33.6**	69.2	99.8
ours3	pseudo w/ aug	✓	✓	44.3	30.7	21.9	40.8	28.1	48.2	**43.8**	27.6	44.5	39.0	**68.3**	**94.9**
whole1	pseudo			50.7	33.3	**21.1**	63.5	26.4	88.6	46.8	26.0	41.0	56.2	71.9	82.8
whole2	pseudo		✓	45.5	**29.4**	21.5	48.5	**23.6**	78.5	**42.9**	**24.8**	**39.7**	36.7	69.4	85.4
whole3	pseudo w/ aug		✓	**41.7**	30.5	21.9	**37.7**	25.1	**57.6**	**42.9**	25.6	48.3	**35.8**	**64.6**	**69.1**
whole4	pseudo w/ aug			48.2	34.4	24.0	44.8	27.7	82.4	47.0	28.6	52.5	44.6	70.1	74.1
sup1	true	✓		**22.4**	14.6	**8.2**	**16.6**	**15.1**	**38.7**	40.7	**3.2**	**12.5**	24.0	**44.3**	**28.6**
sup2	true	✓	✓	**22.4**	**9.4**	8.7	17.2	15.7	41.3	**40.6**	4.0	13.1	**22.8**	44.6	29.6
topline	true		✓	20.5	9.1	7.5	14.6	14.6	38.4	38.7	3.1	12.6	21.9	39.6	25.5

Table 2: *CDER of all systems on each d_{test} at the end of the training sequence, averaged over 10 runs to limit the effect of randomness.*

use generative models to produce synthetic data that mimics past inputs, but generating such realistic conversations on-the-fly is costly and generative models are difficult to train.

As depicted in Figure 2, given an initial model m_0 pre-trained on an out-of-domain corpus (in our case AMI [20]), our goal is to improve overall performance on a *single* target domain d (in our case one of the 11 DIHARD III [15] domains). Since we want to avoid storing potentially sensitive data, we train on one conversation conv_t of d at a time. Hence, any data from steps $< t$ are inaccessible. We believe the combination of augmentation and the stopping criterion described earlier may prevent in-domain forgetting by discouraging overfitting to any single conversation.

4. Experiments

4.1. Dataset

We experiment on *DIHARD III* [15], which contains conversations from 11 different domains shown in Table 1. These domains differ greatly in number of speakers and difficulty (of which diarization error rate is a good proxy). Notice that *webvideo* may not in fact qualify as a domain, as it is a collection of English and Mandarin audio from video sharing platforms. Indeed, its set of shared properties among recordings may be rather small (*e.g.* differences in quality, noise, language, etc.).

4.2. Evaluation metric

Since we want to evaluate the overall quality of a local speaker diarization system that takes 5s inputs, instead of calculating the usual diarization error rate (DER) of a single hypothesis for an entire conversation, we calculate what we name the CDER, *i.e.* the average 5s chunk DER using a 500ms shift. We use `pyannote.audio` [21] to evaluate without forgiveness collar and including all overlapping speech.

4.3. Experimental protocol

We want to determine the best hyper-parameters while still being able to obtain performance for all 11 domains. Hence, we cross-validate hyper-parameter optimization making sure not to leak target-domain knowledge neither in model weights nor in hyper-parameters. We split the DIHARD III *Full* [15] domains into sets DH_A and DH_B as shown in Table 1. Given the differences in domain difficulty, we balance DH_A and DH_B by evening the VB-HMM baseline (track 2) [15] performance between both sets. Our goal is to progressively adapt model m_0 to a *single* domain, so for every hyper-parameter configuration h and every domain $d \in \mathrm{DH}_A$, we train m_0 sequentially on d_{train} and evaluate the resulting model on d_{test} to obtain its CDER p_d. Performance for configuration h is defined as:

$$p_h = \frac{1}{|\mathrm{DH}_A|} \sum_{d \in \mathrm{DH}_A} p_d \tag{3}$$

The configuration with the lowest p_h is used to sequentially train m_0 on each domain $d \in \mathrm{DH}_B$ on d_{train} and evaluating each resulting model on its corresponding d_{test}. The same process is repeated inverting the roles of DH_A and DH_B.

4.4. Implementation details

We use the architecture introduced in [10] (SincNet [22] trainable feature extraction, 4 LSTM [23] and 2 fully-connected layers) that we pre-train with true labels on the AMI corpus [20] training set from *Full* [24], achieving a DER of 17.5 on its test set. This constitutes our initial model m_0. Both strong and weak augmentations aug and Aug apply random noise, but only Aug has a 50% chance of applying a random room impulse response (making it stronger than aug). Noises are sampled from MUSAN [25] (excluding speech) and impulse responses are sampled from *EchoThief* and [26]. We use a separate Adam optimizer [27] for each new conversation and training on train_t is stopped after 3 epochs of no improvement on dev_t. Training sequences are sorted according to existing recording identifiers.

Optimized hyper-parameters are background noise SNR (among ranges 0dB-5dB, 5db-10dB and 10dB-15dB), learning rates (among 10^{-3}, 10^{-4} and 10^{-5}) and batch size (among 16, 32, 64 and 128).

5. Results and discussion

Table 2 summarizes our main results. We include supervised (*sup*) systems as toplines. System *ours3* corresponds to our full version as described in Section 3. The remaining systems constitute various ablative studies. Self-supervised training on the whole target domain at once (*whole*) are non-continual versions of our approach using m_0 to generate pseudo-labels, while *ours1* and *ours2* are ablations of *ours3*.

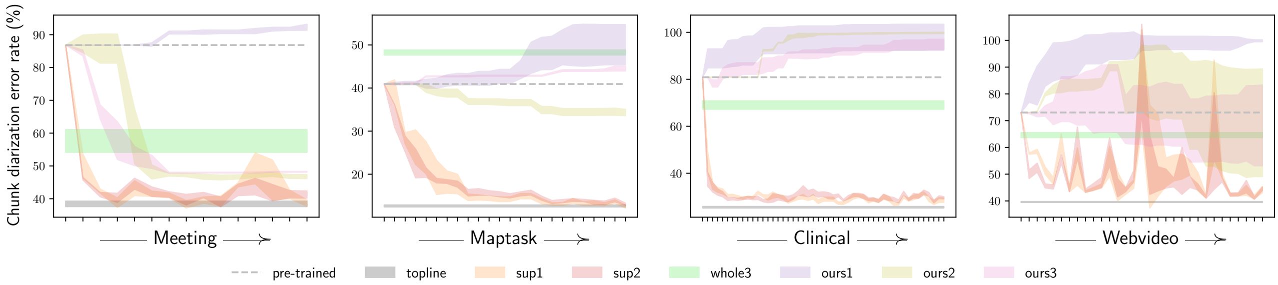

Figure 3: *CDER on d_{test} as a function of training conversations conv_t in sequence. Curves follow the average and standard deviation across 10 runs. Each system is referenced with its identifier from Table 2.*

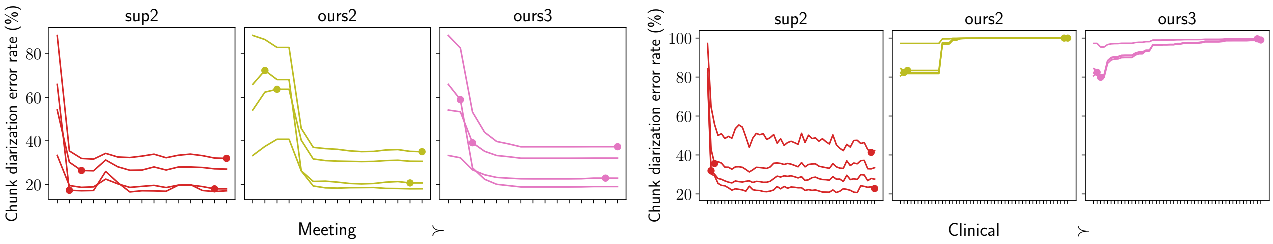

Figure 4: *CDER on the first and last two training conversations. Each curve represents the CDER (averaged over 10 runs) of a single conversation across the entire training sequence. A dot denotes the position of the given conversation within the sequence.*

Continual self-supervision. System *ours2* outperforms *pre-trained* across all domains with a relative improvement of 17% on the average CDER, except on *clinical* where the model diverges. A surprising result from Table 2 is that *ours2* closely follows our best non-continual system *whole3* on average, and even outperforms it in some domains like *meeting*, *maptask* and *socio field*. This suggests that the quality of pseudo-labels may be improving in these domains as new conversations appear, since pseudo-labels in *whole* systems cannot improve because the teacher m_0 is static. Moreover, average CDER between *ours2* and *whole3* only differ by an absolute 1%. Notice that *clinical* for *whole2* (the *whole* equivalent of *ours2*) is the only domain whose performance deteriorates with respect to *pre-trained*. This leads us to believe that divergence may be caused by poor initial pseudo-labels that fail to provide useful information for the model to exploit, effectively reinforcing its own errors as new conversations appear. Finally, note that supervised continual training also performs well, rivaling the non-continual supervised topline with a CDER absolute difference of 2%.

Performance evolution. Figure 3 shows the d_{test} CDER at each conv_t for various domains. System *ours2* seems to keep improving with new conversations, except on *clinical*, where it diverges, and on *webvideo*, where performance of all systems is rather unstable. As discussed before, we believe this may be caused by its loose definition as a domain, suggesting that a well-defined set of shared characteristics may be key to benefit from transfer between conversations.

Forgetting and transfer. Figure 4 shows the train CDER on the first and last two conv_t for domains *meeting* and *clinical*. Notice that performance across conversations varies greatly, suggesting a certain variability in difficulty even in conversations from the same domain. Despite this, forgetting seems to be limited, as sharp increases in CDER are rare when there is no divergence. Notice that overall performance tends to improve with new conversations, which is interesting given that true labels are never seen by *ours2* and *ours3*. Overall, self-supervised systems seem to benefit more from both forward and backward transfer (see CDER before and after the dot in Figure 4 respectively). We believe this may progressively improve pseudo-labels as well as model quality estimation for the stopping criterion. It may also explain performance fluctuations in *webvideo*, as very dissimilar conversations might limit transfer.

The role of augmentation. Our results show that augmentation Aug is key in achieving good self-supervised performance, although not so much in supervised systems. The example of *maptask* in Figure 3 is particularly interesting, as Aug makes the difference between learning and diverging. On the other hand, aug seems to be more useful in *whole* systems than in continual training. Nevertheless, we believe that aug may prevent reinforcing errors at the beginning of continual training when pseudo-label quality is low, although failing to prevent divergence. Figure 4 is a good example of this, as *ours3* is better than *ours2* in the beginning for both *meeting* and *clinical*. Applying aug only during the first conversations of the sequence might be a better strategy to get the best from both variants.

6. Conclusion

We have proposed a training scheme for domain adaptation in end-to-end speaker diarization. We train on the target domain as conversations become available and in a fully self-supervised way, removing the need for annotating data and storing sensitive user conversations permanently. We achieve an average 17% relative improvement over a pre-trained baseline, even rivaling a non-continual self-supervised topline. Moreover, our approach can run locally and autonomously in the background with little to no human involvement (*e.g.* in a home voice assistant).

7. Acknowledgements

This work was granted access to the HPC resources of IDRIS under the allocation AD011012177R1 made by GENCI.

8. References

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

[2] 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 *Interspeech 2020*, 2020, pp. 269–273.

[3] D.-H. Lee *et al.*, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on Challenges in Representation Learning, ICML*, vol. 3, no. 2, 2013, p. 896.

[4] J. Kahn, A. Lee, and A. Hannun, "Self-Training for End-to-End Speech Recognition," in *ICASSP 2020*, 2020, pp. 7084–7088.

[5] Y. Takashima, Y. Fujita, S. Horiguchi, S. Watanabe, P. Garc'ia, and K. Nagamatsu, "Semi-Supervised Training with Pseudo-Labeling for End-to-End Neural Diarization," in *Interspeech 2021*, 2021.

[6] E. Tzinis, Y. Adi, V. K. Ithapu, B. Xu, and A. Kumar, "Continual self-training with bootstrapped remixing for speech enhancement," *arXiv preprint arXiv:2110.10103*, 2021.

[7] R. Hadsell, D. Rao, A. A. Rusu, and R. Pascanu, "Embracing Change: Continual Learning in Deep Neural Networks," *Trends in Cognitive Sciences*, vol. 24, pp. 1028–1040, 2020.

[8] D. Lopez-Paz and M. A. Ranzato, "Gradient Episodic Memory for Continual Learning," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[9] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends in Cognitive Sciences*, vol. 3, no. 4, pp. 128 – 135, 1999.

[10] H. Bredin and A. Laurent, "End-to-end speaker segmentation for overlap-aware resegmentation," in *Interspeech 2021*, 2021.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186.

[12] Y.-A. Chung and J. Glass, "Generative Pre-Training for Speech with Autoregressive Predictive Coding," in *ICASSP 2020*, 2020, pp. 3497–3501.

[13] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A Simple Framework for Contrastive Learning of Visual Representations," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 1597–1607.

[14] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.

[15] N. Ryant, P. Singh, V. Krishnamohan, R. Varma, K. Church, C. Cieri, J. Du, S. Ganapathy, and M. Liberman, "The Third DIHARD Diarization Challenge," *arXiv preprint arXiv:2012.01477*, 2020.

[16] A. Robins, "Catastrophic Forgetting, Rehearsal and Pseudorehearsal," *Connection Science*, vol. 7, no. 2, pp. 123–146, 1995.

[17] L. Pellegrini, G. Graffieti, V. Lomonaco, and D. Maltoni, "Latent Replay for Real-Time Continual Learning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 10 203–10 209.

[18] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual Learning with Deep Generative Replay," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 2990–2999.

[19] D. Rao, F. Visin, A. Rusu, R. Pascanu, Y. W. Teh, and R. Hadsell, "Continual Unsupervised Representation Learning," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 7645–7655.

[20] J. Carletta, "Unleashing the killer corpus: experiences in creating the multi-everything AMI Meeting Corpus," *Language Resources and Evaluation*, vol. 41, no. 2, pp. 181–190, 2007.

[21] H. Bredin, R. Yin, J. M. Coria, G. Gelly, P. Korshunov, M. Lavechin, D. Fustes, H. Titeux, W. Bouaziz, and M.-P. Gill, "pyannote.audio: neural building blocks for speaker diarization," in *ICASSP 2020*, Barcelona, Spain, May 2020.

[22] M. Ravanelli and Y. Bengio, "Speaker Recognition from Raw Waveform with SincNet," *2018 IEEE Spoken Language Technology Workshop (SLT)*, pp. 1021–1028, 2018.

[23] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

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

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

[26] J. Traer and J. H. McDermott, "Statistics of natural reverberation enable perceptual separation of sound and space," *Proceedings of the National Academy of Sciences*, vol. 113, no. 48, pp. E7856–E7865, 2016.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.