



Powerset multi-class cross entropy loss for neural speaker diarization

Alexis Plaquet & Hervé Bredin

IRIT, Université de Toulouse, CNRS, Toulouse INP, UT3, Toulouse, France

firstname.lastname@irit.fr

Abstract

Since its introduction in 2019, the whole end-to-end neural diarization (EEND) line of work has been addressing speaker diarization as a frame-wise multi-label classification problem with permutation-invariant training. Despite EEND showing great promise, a few recent works took a step back and studied the possible combination of (local) supervised EEND diarization with (global) unsupervised clustering. Yet, these hybrid contributions did not question the original multi-label formulation. We propose to switch from multi-label (where any two speakers can be active at the same time) to powerset multi-class classification (where dedicated classes are assigned to pairs of overlapping speakers). Through extensive experiments on 9 different benchmarks, we show that this formulation leads to significantly better performance (mostly on overlapping speech) and robustness to domain mismatch, while eliminating the detection threshold hyperparameter, critical for the multi-label formulation.

Index Terms: speaker diarization, loss function, powerset classification

1. Introduction

Speaker diarization is the task of partitioning an audio stream into homogeneous temporal segments according to the identity of the speaker. Most dependable diarization approaches consist of a cascade of several steps [1]: voice activity detection to discard non-speech regions, speaker embedding to obtain discriminative speaker representations, and clustering to group speech segments by speaker identity. This family of approaches has two main drawbacks:

- errors made by each step are propagated to the subsequent steps, possibly escalating into even larger errors;
- they are not designed to detect (let alone assign to the right speakers) overlapping speech regions – an additional post-processing step is needed to handle them [2].

To circumvent these limitations, a new family of approaches have recently emerged, rethinking speaker diarization completely [3, 4, 5]. Dubbed end-to-end diarization (EEND), the main idea is to train a single neural network that ingests the audio recording and directly outputs the diarization, hence addressing the error propagation issue. They formulate the problem as a *multi-label classification* task, allowing multiple overlapping speakers to be active simultaneously. Furthermore, *permutation-invariant training* [3] is the critical ingredient that turns clustering – an intrinsically unsupervised task – into a supervised classification task. EEND approaches do have a few significant limitations:

- they struggle to predict the correct number of speakers, espe-

cially when it is larger in test data than during training – additional mechanisms such as encoder-decoder attractors were proposed to partially solve this issue [6];

- training such approaches is extremely data hungry as each conversation in the training set constitutes a single training sample – one has to resort to synthetic (hence unrealistic) conversations instead [7];
- because of the internal self-attention mechanism, they do not scale well to long conversations.

Therefore, a few recent works took a step back to borrow the *best of both worlds* (BoBW, *i.e.* multi-stage and end-to-end approaches) [8, 9]. The general principle is based on three steps:

1. split long conversations into shorter chunks;
2. apply end-to-end speaker diarization on each of them;
3. stitch them back together using speaker embeddings and unsupervised clustering.

To both start with a strong baseline and ensure reproducibility, we build on top of the speaker diarization pipeline available in version 2.1.1 of `pyannote.audio` open source toolkit [10] that embraces this aforementioned three-steps principle. More precisely, we focus on the loss function used for training neural networks towards end-to-end speaker diarization of short audio chunks. [5] calls this task *speaker segmentation* and states that working with short audio chunks has the following advantages:

- it circumvents the scalability issue because the neural network only ever ingests short fixed-duration (5s in our case) audio chunks;
- it makes it easier to train (as each conversation can now provide multiple training samples);
- it allows to set a small upper bound on the (local) number of speakers K_{\max} – for instance, there is a 99% chance that any 5s chunk contains less than $K_{\max} = 3$ speakers in the 9 benchmarking datasets used in this paper.

Since its introduction a few years ago, the EEND line of work has been processing whole conversations with up to $K_{\max} \approx 20$ (partially overlapping) speakers. Therefore, addressing it as a frame-wise multi-label classification problem was a brilliant (and efficient!) idea that we summarize in Section 2. Working with a much smaller number of ($K_{\max} = 3$) speakers opens up a new set of opportunities which have not yet been investigated by recent BoBW approaches. Inspired by [11], we propose to address *speaker segmentation* as a frame-wise *powerset* multi-class classification problem. Section 3 provides a detailed description of the approach but one can understand the main idea by looking at Figure 1. The advantages of this approach are two-fold: it completely removes the need for the (sensitive) detection threshold hyperparameter, and it leads to significant performance improvement of the overall

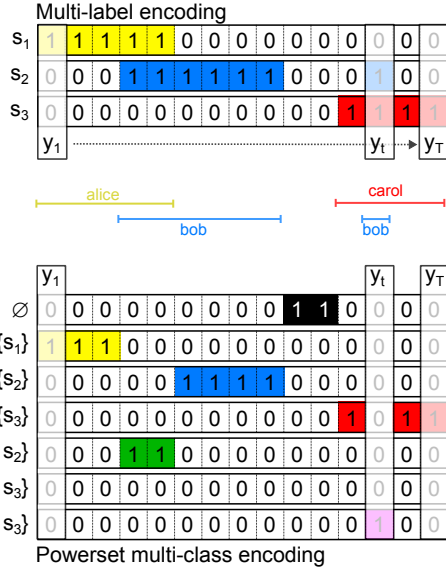


Figure 1: From multi-label to powerset multi-class encoding. While powerset classes are mutually exclusive, several multi-label classes must be active at the same time to encode overlapping speakers.

speaker diarization pipeline (as shown by a thorough evaluation on 9 benchmarking datasets in Sections 4 and 5). In particular, we claim state-of-the-art performance on AISHELL-4 [12], AliMeeting [13], AMI [14], Ego4D [15], MSDWild [16], and REPERE [17] benchmarks, make the approach available in the `pyannotate.audio` library, and release pretrained models publicly.

2. From multi-label classification...

As depicted in the upper part of Figure 1, the reference segmentation of an audio chunk \mathbf{X} can be encoded into a sequence of K_{\max} -dimensional binary frames $\mathbf{y} = \{\mathbf{y}_1, \dots, \mathbf{y}_T\}$ where $\mathbf{y}_t \in \{0, 1\}^{K_{\max}}$ and $y_t^k = 1$ if speaker k is active at frame t and $y_t^k = 0$ otherwise. We may arbitrarily sort speakers by chronological order of their first activity but any permutation of the K_{\max} dimensions is a valid representation of the reference segmentation. Therefore, the binary cross entropy loss function \mathcal{L}_{BCE} (classically used for such multi-label classification problems) has to be turned into a permutation-invariant loss function \mathcal{L} by running over all possible permutations $\text{perm}(\mathbf{y})$ of \mathbf{y} over its K_{\max} dimensions:

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = \min_{\text{perm}(\mathbf{y})} \mathcal{L}_{\text{BCE}}(\text{perm}(\mathbf{y}), \hat{\mathbf{y}}) \quad (1)$$

with $\hat{\mathbf{y}} = f(\mathbf{X})$ where f is the segmentation model whose architecture is described below. In practice, for efficiency, we first compute the $K_{\max} \times K_{\max}$ binary cross entropy losses between all pairs of \mathbf{y} and $\hat{\mathbf{y}}$ dimensions, and rely on the Hungarian algorithm to find the permutation that minimizes the overall binary cross entropy loss.

For the segmentation model f , we use the exact same architecture as the one introduced by [5]. It consists in a *SincNet* convolutional block [18] (that performs frame-wise feature extraction), four bi-directional LSTMs (that contains most of the learnable weights), two fully-connected layers, a classification layer with K_{\max} outputs, and a final *sigmoid* activation function to squash the frame-wise speaker activities $\hat{\mathbf{y}}$ between 0 and 1.

At test time, a subsequent binarization step is needed to decide whether each speaker s is active in each frame t . This is achieved by comparing \hat{y}_{st} to a detection threshold $\theta \in [0, 1]$ whose value needs to be tuned carefully.

3. ... to powerset multi-class classification

While the standard multi-label loss relies on $K_{\max} = 3$ classes (one for each speaker s_1, s_2, s_3) that may be active simultaneously to encode overlapping speech, the *powerset* multi-class encoding considers a grand total of $K_{\text{powerset}} = 7$ mutually exclusive classes:

- \emptyset for non-speech frames;
- $\{s_1\}, \{s_2\}, \{s_3\}$ for frames with one active speaker;
- $\{s_1, s_2\}, \{s_1, s_3\}, \{s_2, s_3\}$ for two overlapping speakers¹.

Switching to *powerset* encoding entails no radical change to the rest of the approach. It only amounts to changing the output size of the classification layer from $K_{\max} = 3$ to $K_{\text{powerset}} = 7$, the final activation function from *sigmoid* to *softmax*, and the training loss function from binary to regular cross-entropy. Furthermore, a nice by-product is that the critical detection threshold θ is removed in favor of a simple *argmax* at test time.

Yet, *powerset* permutation space is slightly more complex than in the multi-label setting. For example, swapping speakers s_1 and s_2 in the multi-label case obviously implies swapping classes $\{s_1\}$ and $\{s_2\}$, but also overlapping speakers classes $\{s_1, s_3\}$ and $\{s_2, s_3\}$. Therefore, to find the optimal permutation in the *powerset* space, we use the following process:

- convert target (\mathbf{y}) from *powerset* to multi-label encoding;
- convert binarized prediction ($\arg \max_k \hat{\mathbf{y}}$) from *powerset* to multi-label encoding;
- find the optimal permutation as described in Section 2;
- permute the target accordingly in the multi-label space, and convert it back to *powerset* encoding;
- compute the cross-entropy loss in the *powerset* space.

While [11] introduced the use of *powerset* encoding for speaker diarization, it has two main limitations. First, *powerset* encoding does not play well with a variable number of speakers: adding an extra speaker to a pool of K speakers results in $K + 1$ new *powerset* classes (one for the actual speaker, and one for each possible overlapping speakers classes). Therefore, [11] had to resort to use a fixed and conservatively high number of speakers ($K_{\max} = 16$). Second, this leads to a number of *powerset* classes which is prohibitively high. For instance, using $K_{\max} = 16$ results in $K_{\text{powerset}} = \binom{0}{K} + \binom{1}{K} + \binom{2}{K} = 137$ classes, most of which are almost never seen during training. For reasons already explained in the introduction, our approach overcomes both issues: we can stick to a small upper bound on the number of speakers ($K_{\max} = 3$) thanks to our processing of short audio chunks, and the unsupervised clustering step takes care of estimating the number of speakers specific to each conversation later in the pipeline.

4. Experiments

To ensure reproducibility, the research described in this paper builds on top of the speaker diarization pipeline from version 2.1.1 of `pyannotate.audio` open source toolkit that follows the

¹We do not consider the case of three or more overlapping speakers because the number of frames for which this happens in our benchmarking datasets is marginal: 1.6% in the compound dataset, 0.73% in DIHARD III.

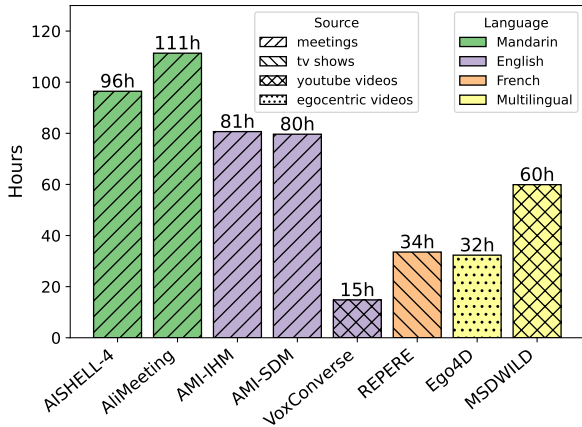


Figure 2: Compound multi-domain training set.

three-steps principle of BoBW approaches. More details on the pipeline internals are available in [10] but we describe the gist of it in the following section.

4.1. Baseline

The first step consists in applying the pretrained end-to-end neural speaker segmentation model `pyannote/segmentation` introduced in [5] using a sliding window of 5s with a step of 500ms. A binarization step is further applied using the detection threshold $\theta \in [0, 1]$ which constitutes the first hyperparameter of the approach. The second step consists in extracting one speaker embedding per active speaker in each 5s window. More precisely, speaker embeddings are only extracted from audio samples with exactly one active speaker: overlapping speech regions inferred automatically from the first step are discarded before computing the embeddings. All experiments reported in this paper rely on the pretrained ECAPA-TDNN model [19] from SpeechBrain [20] available at hf.co/speechbrain/spkrec-ecapa-voxceleb. The third and final step consists in applying agglomerative clustering on the aforementioned embeddings, using a second hyperparameter δ as the maximum allowed distance between centroids of two clusters for them to be merged. Once each local (*i.e.* from first step) active speaker is assigned to a global cluster, the global speaker diarization output can be constructed.

4.2. Datasets

We report performance on the official test sets of 9 different datasets: AISHELL-4 [12], AliMeeting [13], AMI [14] (with two variants), DIHARD III [21], Ego4D² [15], MSDWild [16], REPERE [17], and VoxConverse [22]. AMI variants are *headset mix* (or *IHM*, merged audio of the participants’ headset microphones) and *array 1 channel 1* (or *SDM*, first channel from the first far-field microphone array).

For datasets distributed with an official training/development split, we use the training subset as is. For those only providing a development set, we split it into two parts which we call training and development subsets in the rest of the paper. Models were trained on a compound

²At the time of writing this paper, test labels of Ego4D were not available. Hence, Ego4D results were obtained on its *development* subset and should therefore be taken with a grain of salt.

multi-domain training set made of the concatenation of the training sets of all aforementioned datasets, but DIHARD. DIHARD was kept aside to measure how robust our models are to domain mismatch: it is composed of 11 sub-domains, all tested individually in this paper. Figure 2 showcases the amount of data from each dataset in the compound training set as well as their source and language. We do expect that including DIHARD in the compound training set would significantly improve the results found in this paper.

While Figure 2 highlights the relative imbalance between each dataset in the compound training set, we do make sure that the compound development set is balanced, in two variants:

- `dev_duration` is built by picking random 5s chunks from the *development* subset of each dataset, for a grand total of 8 hours (one hour per dataset);
- `dev_files` is built by randomly picking 25 files from the *development* subset of each dataset³, for a grand total of 164 files.

4.3. Experimental protocol

Speaker segmentation models are trained on the compound training set for at most one hundred hours, using Adam optimizer, an initial learning rate of 10^{-3} , and a scheduler that divides the learning rate by 2 after 30 epochs with no improvement. The validation metric used by the scheduler and for model selection is the local (*i.e.* computed on 5s chunks) diarization error rate on the `dev_duration` development set.

hyperparameters tuning is performed with `pyannote.pipeline` in order to minimize the average diarization error rate on the `dev_files` compound development subset. More specifically, in the case of an internal multi-label speaker segmentation model, two hyperparameters need to be optimized: the detection threshold θ and the clustering threshold δ . In the case of *powerset* segmentation models, the detection threshold θ is not needed (because it is replaced by a parameter-less *argmax*): only the clustering threshold δ needs to be optimized.

Domain adaptation experiments reported in Section 5 are done by fine-tuning speaker segmentation models further on one specific domain. In those cases, for each dataset (or domain in case of DIHARD), model fine-tuning relies on the corresponding training set, model selection relies on the whole development set, and so does the tuning of pipeline hyperparameters.

5. Results and discussions

While the first line of Table 1 reports the performance of the speaker diarization pipeline based on the pretrained multi-label model available at hf.co/pyannote/segmentation, the second line is our (successful) attempt at retraining it from scratch with the proposed compound training set.

Powerset compound training consistently gets better results than its multi-label counterpart – with an average relative improvement of 8% for *in domain* datasets (from 25.6% to 23.5% diarization error rate). Further adapting the *powerset* model to each dataset gives an additional 8% relative improvement, reaching state of the art performance for most of them.

Both approaches are tested for robustness to domain mismatch using the DIHARD III dataset, which is composed of 11 domains that were never seen during (compound) train-

³Except for AISHELL-4, AliMeeting, and AMI which have only 20, 8, and 18 files in their respective development subsets.

Table 1: Performance on 8 different datasets. We report diarization error rates on the official test sets (with the exception of Ego4D validation set because test labels are not available). **Best result for each dataset** are reported in bold (as well as those less than 5% worse relatively). No forgiveness collar is used for evaluation, except for numbers in italics with a grey background computed with a 250ms forgiveness collar (to allow comparison with some results reported in the literature). 🏆 indicates the winning (ensemble) submission to the VoxSRC or DIHARD III challenges.

	AISHELL-4 channel 1 [12]	AliMeeting channel 1 [13]	AMI headset mix [14]	AMI array 1 channel 1 [14]	Ego4D v1 validation [15]	MSDWild [16]	REPERE phase 2 [17]	VoxConverse v0.3 [22]	in domain average	DIHARD III [21]
Pretrained pyannote/segmentation baseline [5]	14.7	26.1	19.0	29.1	63.2	34.8	8.5	11.9	25.9	23.7
Multi-label compound training	17.1	27.2	22.3	24.7	58.7	33.5	9.0	12.1	25.6	33.8
↳ with domain adaptation	14.0	25.7	19.8	24.5	52.4	27.1	9.1	11.3	23.0	22.4
↳ with 250ms forgiveness collar	<i>8.3</i>	<i>15.7</i>	<i>12.2</i>	<i>16.3</i>	<i>43.0</i>	16.0	6.9	<i>6.4</i>	<i>15.6</i>	<i>12.3</i>
Powerset compound training	16.9	23.3	19.7	22.0	57.2	29.2	8.4	11.6	23.5	29.9
↳ with domain adaptation	13.2	24.5	18.0	22.9	48.2	28.5	8.2	10.4	21.7	21.3
↳ with 250ms forgiveness collar	<i>7.6</i>	<i>15.8</i>	<i>11.2</i>	<i>15.3</i>	<i>38.9</i>	<i>18.6</i>	6.3	<i>5.8</i>	14.9	<i>12.1</i>
State of the art as of Feb. 2023 [Source]	16.1 [23]	23.5 [23]	19.0 [1]	23.7 [23]	67.2 [15]	22.0 [16]	12.6 [24]	5.1 🏆		16.8 🏆

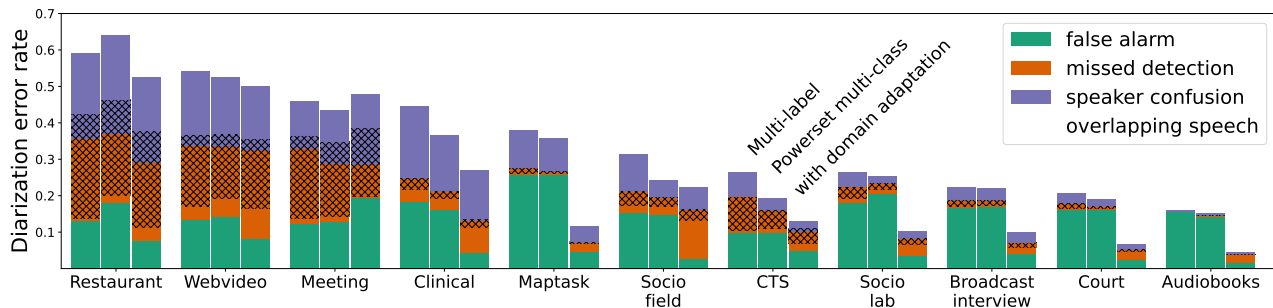


Figure 3: Decomposition of diarization error rate on DIHARD III in terms of its false alarm, missed detection, and speaker confusion components. Errors happening in conjunction with overlapping speakers are hatched.

ing. Powerset training shows a 11% relative improvement over multi-label (from 33.8% to 29.9%), even larger than for the *in domain* case. This suggests that powerset training is more robust to domain mismatch than its multi-label variant – a behavior that could be explained by the removal of the otherwise sensitive detection threshold θ .

Figure 3 provides a detailed analysis of the errors committed by the various approaches. We find that most of the improvement comes from a consistent reduction in missed overlapping speech. This also holds for *in-domain* data (with missed detection rate reduced from 13.1% to 9.9% on average), supporting the idea that powerset explicit modeling of overlapping speakers classes has a huge impact on the overall performance.

Fine-tuning the powerset segmentation model on each DIHARD domain (*‘with domain adaptation’* bars in Figure 3) tends to converge towards an overall better compromise between false alarm and missed detection rates, with little to no impact on speaker confusion. This suggests that this additional domain adaptation step also plays the role of the detection threshold θ (which is a separate hyperparameter in the multi-label case); thus making the whole approach even more robust.

6. Conclusion

In this paper, we study the impact of framing the speaker diarization task as a powerset multi-class classification problem. We extensively test this approach on 8 *in domain* datasets, as well as the 11 domains of DIHARD used as *out-of-domain* data. Compared to the classic multi-label approach, using a powerset multi-class encoding results in significant diarization error rate improvement (mostly due to better predictions on overlapping speech) and better robustness to domain mismatch. We obtain state of the art performance on AISHELL-4, AliMeeting, AMI, Ego4D, MSDWild, and REPERE. In the spirit of reproducible research, the powerset multi-class segmentation code is available in the open-source `pyannote.audio` library. The models trained on the compound training dataset, as well as their recomputed output on each separate dataset are available at github.com/FrenchKrab/IS2023-powerset-diarization.

7. Acknowledgements

This work was granted access to the HPC resources of IDRIS under the allocations AD011013477 and AD011012177R2 made by GENCI.

8. References

- [1] 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*, 2022.
- [2] L. Bullock, H. Bredin, and L. P. Garcia-Perera, “Overlap-aware diarization: Resegmentation using neural end-to-end overlapped speech detection,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7114–7118.
- [3] Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, and S. Watanabe, “End-to-End Neural Speaker Diarization with Permutation-free Objectives,” in *Interspeech*, 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] H. Bredin and A. Laurent, “End-to-end speaker segmentation for overlap-aware resegmentation,” in *Proc. Interspeech 2021*, 2021.
- [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] F. Landini, A. Lozano-Diez, M. Diez, and L. Burget, “From Simulated Mixtures to Simulated Conversations as Training Data for End-to-End Neural Diarization,” in *Proc. Interspeech 2022*, 2022, pp. 5095–5099.
- [8] 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*, 2021.
- [9] —, “Advances in Integration of End-to-End Neural and Clustering-Based Diarization for Real Conversational Speech,” in *Proc. Interspeech 2021*, 2021.
- [10] H. Bredin, “pyannote.audio 2.1 speaker diarization pipeline: principle, benchmark, and recipe,” in *Proc. Interspeech 2023*, 2023.
- [11] Z. Du, S. Zhang, S. Zheng, and Z.-J. Yan, “Speaker overlap-aware neural diarization for multi-party meeting analysis,” in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 7458–7469. [Online]. Available: <https://aclanthology.org/2022.emnlp-main.505>
- [12] Y. Fu, L. Cheng, S. Lv, Y. Jv, Y. Kong, Z. Chen, Y. Hu, L. Xie, J. Wu, H. Bu, X. Xu, J. Du, and J. Chen, “AISHELL-4: An Open Source Dataset for Speech Enhancement, Separation, Recognition and Speaker Diarization in Conference Scenario,” in *Proc. Interspeech 2021*, 2021.
- [13] F. Yu, S. Zhang, Y. Fu, L. Xie, S. Zheng, Z. Du, W. Huang, P. Guo, Z. Yan, B. Ma, X. Xu, and H. Bu, “M2MeT: The ICASSP 2022 Multi-Channel Multi-Party Meeting Transcription Challenge,” in *Proc. ICASSP 2022*, 2022.
- [14] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, and M. Kronenthal, “The AMI Meetings Corpus,” in *Proc. Symposium on Annotating and Measuring Meeting Behavior*, 2005.
- [15] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, M. Martin, T. Nagarajan, I. Radosavovic, S. K. Ramakrishnan, F. Ryan, J. Sharma, M. Wray, M. Xu, E. Z. Xu, C. Zhao, S. Bansal, D. Batra, V. Cartillier, S. Crane, T. Do, M. Doulaty, A. Erapalli, C. Feichtenhofer, A. Fragomeni, Q. Fu, C. Fuegen, A. Gebreselasie, C. Gonzalez, J. Hillis, X. Huang, Y. Huang, W. Jia, W. Khoo, J. Kolar, S. Kottur, A. Kumar, F. Landini, C. Li, Y. Li, Z. Li, K. Mangalam, R. Modhugu, J. Munro, T. Murrell, T. Nishiyasu, W. Price, P. R. Puentes, M. Ramazanov, L. Sari, K. Somasundaram, A. Southerland, Y. Sugano, R. Tao, M. Vo, Y. Wang, X. Wu, T. Yagi, Y. Zhu, P. Arbelaez, D. Crandall, D. Damen, G. M. Farinella, B. Ghanem, V. K. Ithapu, C. V. Jawahar, H. Joo, K. Kitani, H. Li, R. Newcombe, A. Oliva, H. S. Park, J. M. Rehg, Y. Sato, J. Shi, M. Z. Shou, A. Torralba, L. Torresani, M. Yan, and J. Malik, “Ego4D: Around the World in 3,000 Hours of Egocentric Video,” in *Proc. CVPR 2022*, 2022.
- [16] T. Liu, S. Fan, X. Xiang, H. Song, S. Lin, J. Sun, T. Han, S. Chen, B. Yao, S. Liu, Y. Wu, Y. Qian, and K. Yu, “MSDWild: Multimodal Speaker Diarization Dataset in the Wild,” in *Proc. Interspeech 2022*, 2022, pp. 1476–1480.
- [17] J. Kahn, O. Galibert, L. Quintard, M. Carré, A. Giraudel, and P. Joly, “A Presentation of the REPERE Challenge,” in *Proc. CBMI 2012*, 2012.
- [18] M. Ravanelli and Y. Bengio, “Speaker recognition from raw waveform with sincnet,” in *Proc. SLT 2018*, 2018.
- [19] 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.
- [20] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, J.-C. Chou, S.-L. Yeh, S.-W. Fu, C.-F. Liao, E. Rastorgueva, F. Grondin, W. Aris, H. Na, Y. Gao, R. D. Mori, and Y. Bengio, “SpeechBrain: A General-Purpose Speech Toolkit,” 2021, arXiv:2106.04624.
- [21] N. Ryant, P. Singh, V. Krishnamohan, R. Varma, K. Church, C. Cieri, J. Du, S. Ganapathy, and M. Liberman, “The Third DI-HARD Diarization Challenge,” in *Proc. Interspeech 2021*, 2021, pp. 3570–3574.
- [22] J. S. Chung, J. Huh, A. Nagrani, T. Afouras, and A. Zisserman, “Spot the Conversation: Speaker Diarisation in the Wild,” in *Proc. Interspeech 2020*, 2020.
- [23] D. Raj, D. Povey, and S. Khudanpur, “Gpu-accelerated guided source separation for meeting transcription,” 2022.
- [24] 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 *Proc. ICASSP 2020*, 2020.