Audio-Visual Speech Separation via Bottleneck Iterative Network

Sidong Zhang, Shiv Shankar, Trang Nguyen, Madalina Fiterau
Manning College of Information & Computer Sciences
University of Massachusetts Amherst
Amherst, USA

Andrea Fanelli
Dolby Laboratories
San Francisco, USA

***Abstract*—Integration of information from non-auditory cues can significantly improve the performance of speech-separation models. Often such models use deep modality-specific networks to obtain unimodal features, which are statically combined to obtain high-level fused representations. Such designs often risk being too costly or lightweight but lacking capacity. In this work, we present an iterative representation refinement approach called Bottleneck Iterative Network (BIN), a technique that repeatedly progresses through a lightweight fusion block, while controlling nuisance representations in fusion by using bottleneck fusion tokens. This helps improve the capacity of the model, while avoiding major increase in model size and balancing between the model performance and training cost. We test BIN on challenging noisy audio-visual speech separation tasks, and show that our approach consistently outperforms state-of-the-art benchmark models with respect to SI-SDRi on NTCD-TIMIT and LRS3+WHAM! datasets, while simultaneously reducing the training and GPU inference time by 75%.**

***Index Terms*—Multimodal Speech separation, Neural network**

I. Introduction

The goal of speech separation is to separate a multi-speaker audio stream into multiple single-speaker audio streams. This task is critical not only as a standalone application but also as an important first step in several downstream applications like speech enhancement and audio editing. Pure audio-based methods for this speech separation task have long been studied in signal-processing [6, 15], but often face limitations when factors like reverberations, heavy overlap and background noise are common [18]. Inspired by multimodal speech cognition in humans [20, 17], contemporary research is pivoting towards a multimodal approach that integrates visual cues in the speech separation task. The Audio-Visual Speech Separation (AVSS) problem is about taking a single-channel audio stream of multiple speakers along with a corresponding video that captures all the speakers' faces, with the goal of identifying the utterance of each speaker [15, 14, 16].

Related Work: AVconvTasnet [15] extended the ConvTasNet [26] paradigm, an audio-only speech separation state-of-the-art model, to the audio-visual domain, setting one of the earlier baselines for the AVSS task. Since then, there have been various deep learning methods developed for the task [10, 11]. Gao and Grauman [4] used speaker images as external cues for separating audio using a large foundation model. RTFS-Net [21] currently sets the frontier with respect to speech separation quality but comes with not only expensive training cost but also high inference time. This is a critical drawback because lots of important audio enhancement applications, such as real-time conference call enhancement, require fast speech separation. AVLIT [16] and IIA-Net [13] are recent lightweight AVSS models. However, we find that there is a significant gap in output quality for noisy AVSS between these lightweight models and SOTA (state-of-the-art) RTFS-Net.

To address this gap, we introduce a new AVSS model, called Bottleneck Iterative Network (BIN), that iteratively refines the audio and visual representations using their fused representation via a repetitive progression through the bottleneck fusion variables and the outputs of the two modalities from the same fusion block. Tested on two popular AVSS benchmarks, BIN strikes a good balance between speech separation quality and computing resources, being on par with RTFS-Net's state-of-the-art performance (and even improving on SI-SDR) while saving up to 75% training time and GPU inference time.

Analysis on the progression of speech separation quality and separation mask throughout multiple iterations of fusion indicate the advantage of iterative fusion in the AVSS problem. Furthermore importantly, our approach of iterative fusion refinement is model- and task-agnostic and potentially extensible to a wider variety of multimodal tasks and architectures, which can be further investigated in future works.

II. Bottleneck Iterative Network

The noisy AVSS problem can be formulated as follows. There are two inputs: a noisy audio mixture record $\mathbf{s}$, which is composed of clean utterances of M speakers $\mathbf{s}_{1\ldots M}$ and background noise $\mathbf{n}$, and video input $\mathbf{v}$, which is a concatenation of video frames of lip regions $\mathbf{v}_1$ to $\mathbf{v}_M$ from the M speakers. The expected outputs are a list of M separated single-speaker audio streams $\hat{\mathbf{s}}_{1\ldots M}$. Our proposed BIN (illustrated in Figure 1) for noisy AVSS consists of the following components:

Audio embedding model $\mathcal{E}_A$: For an input audio mixture $\mathbf{s}$ of shape $1 \times T$, where T is the length of the audio times the sampling rate, we work on a latent embedding space $\mathcal{E}_A(\mathbf{s})$ by preprocessing the audio with an 1D convolution network $\mathcal{E}_A$. $\mathcal{E}_A(\mathbf{s})$ embeds $\mathbf{s}$ to a latent space of dimension $C_A \times F_A$, where C_A expands the first dimension of $\mathbf{s}$ to a deeper hidden channel size, and F_A compresses the time axis. $\mathcal{E}_A$ is only called once before the BIN iteration for audio preprocessing.

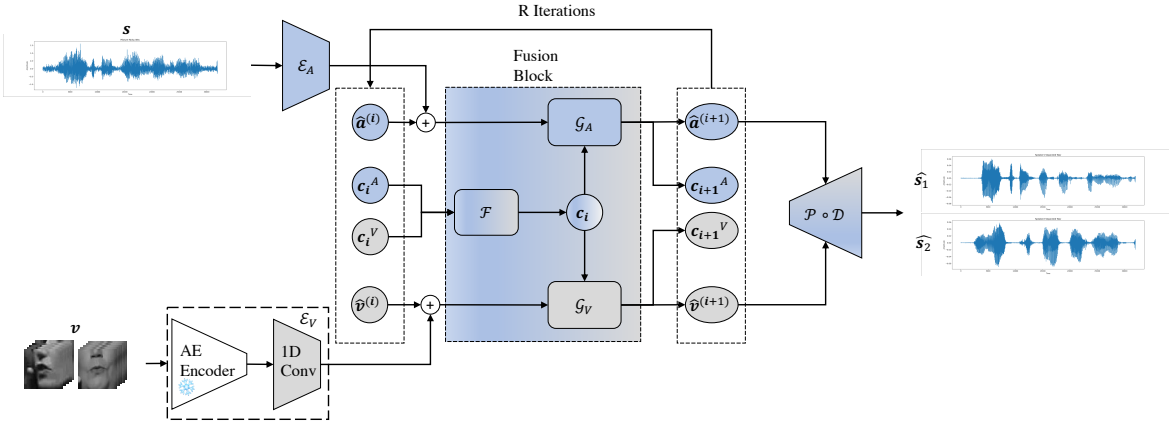

Fig. 1: BIN for Noisy AVSS model design.

Video embedding model $\mathcal{E}_V$: The video embedding model consists of the encoder part of a 4-layer convolutional autoencoder pretrained for the video recover task, and a 1D convolution network. We freeze the autoencoder's parameters during training. A video sample $\mathbf{v}$ is of shape $N \times Fr \times C \times H \times W$, where N is the number of speakers, Fr is the total number of frames, C is the RGB channel size, and $H \times W$ represents the height and width of each frame. The embedding space is of shape $C_V \times F_V$, where F_V preserves a compacted sequential information along the time axis and C_V is the latent dimension combining the number of the speakers, the RGB channel size, height and width for each frame. We further interpolate the video embedding so that $F_V = F_A$.

Fusion Block: As detailed in Figure 1, there are four components in the Fusion Block. The Fusion Block is repeated for R iterations, with shared parameters.

a) Fusion variable: $\mathbf{c}$: We utilize two learnable fusion variables $\mathbf{c}^A$ of shape $C_{AH} \times F_A$ and $\mathbf{c}^V$ of shape $C_{VH} \times F_V$, update them with the partial output of the audio and video feature generator, and apply the variable fusion function on them, as explained later in Equations 1,2,3.

b) Audio feature generator: $\mathcal{G}_A$: At iteration i, $\mathcal{G}_A$ takes the generated audio feature from the previous iteration as the backward connection for refinement with a residual connection to $\mathcal{E}_A(\mathbf{s})$, a fusion variable $\mathbf{c}_{i-1}$ from last iteration for the bottleneck audio-visual fusion information, and outputs the new audio feature $\hat{\mathbf{a}}^{(i)}$ together with $\mathbf{c}_i^A$. Formally,

$$\hat{\mathbf{a}}^{(i)} || \mathbf{c}_i^A = \mathcal{G}_A(\hat{\mathbf{a}}^{(i-1)} + \mathcal{E}_A(\mathbf{s}), \mathbf{c}_{i-1}) \tag{1}$$

We define $\hat{\mathbf{a}}^{(0)} = \mathbf{0}$. At the first iteration, $\mathbf{c}_0$ is defined as the average of the learnable $\mathbf{c}^A$ and $\mathbf{c}^V$. For our model, we used an Asynchronous Fully Recurrent Convolutional Neural Network (A-FRCNN) [7] as the audio feature generator on a convolved concatenation of $\hat{\mathbf{a}}^{(i-1)} + \mathcal{E}_A(\mathbf{s})$ and $\mathbf{c}_{i-1}$.

c) Video feature generator: $\mathcal{G}_V$: We follow the same scheme as in $\mathcal{G}_A$, and produce $\hat{\mathbf{v}}^{(i)}$ together with $\mathbf{c}_i^V$. Mathematically, at iteration i,

$$\hat{\mathbf{v}}^{(i)} || \mathbf{c}_i^V = \mathcal{G}_V(\hat{\mathbf{v}}^{(i-1)} + \mathcal{E}_V(\mathbf{v}), \mathbf{c}_{i-1}) \tag{2}$$

Similar to the audio-modality we initialize $\hat{\mathbf{v}}$ with zeroes , and use an A-FRCNN [7] as the video feature generator

d) Variable Fusion function: $\mathcal{F}$: We choose a simple aggregation based fusion function [9] to reduce compute time and model complexity. For a given $\mathbf{c}_i^A$ and $\mathbf{c}_i^V$,

$$\mathbf{c}_i = \mathcal{F}(\mathbf{c}_i^A, \mathbf{c}_i^V) = \frac{1}{2}(\mathbf{c}_i^A + \mathbf{c}_i^V) \tag{3}$$

Predictive function $\mathcal{P}$ and **audio decoder** $\mathcal{D}$: A common practice to separate audio records in AVSS tasks is to predict the masks for the clean audio records [26, 16, 12]. Given the output $\hat{\mathbf{a}}^{(R)}$ and $\hat{\mathbf{v}}^{(R)}$ after R iterations, a predictive function $\mathcal{P}$ is a 1D convolution network mapping the latent output of A-FRCNNs back to the embedding space as the mask $\mathbf{m}$:

$$\mathbf{m} = \mathcal{P}(\hat{\mathbf{a}}^{(R)}, \hat{\mathbf{v}}^{(R)}) \tag{4}$$

Then the 1D transposed convolution network $\mathcal{D}$ retrieves the clean audio from the masked embedding audio:

$$\hat{\mathbf{s}}_i = \mathcal{D}(\mathcal{E}_A(\mathbf{s}) \odot \mathbf{m})[i-1], i \in \{1, 2, \cdots, M\} \tag{5}$$

where $\odot$ represents the element-wise product.

A major design feature in BIN is that the multimodal fusion relies on only fusing the partial output of $\mathcal{G}_A$ and $\mathcal{G}_V$ using $\mathcal{F}$, $\mathcal{G}_A$, and $\mathcal{G}_V$, bottlenecking the cross-modality information exchange between the encoded modality. The bottleneck is important when certain modality inputs have a significant level of nuisance, as the fusion bottleneck forces the information exchange to happen through a narrow channel, extracting only AVSS relevant information. Bottlenecking layers has been shown effective for regularization and has a compression and distillation effect [23]. Previous research [19] has shown that bottlenecking cross-attention tokens in transformers leads to more robust performance multimodal action recognition tasks.

Following AVLIT [16], we use the lightweight A-FRCNN instead of multiple recurrent networks in RTFS-Net [21]. Unlike AVLIT which directly fuses audio and visual modality, we use the more computationally efficient bottlenecked fusion mechanism. Thus our compute time remains manageable despite having multiple iterations over the fusion blocks.

III. Experiments

A. Datasets

We experiment on two datasets studied in Martel et al. [16]

NTCD-TIMIT: Following Martel et al. [16], we mix clean audio records from TCD-TIMIT dataset [5] with noise sampled uniformly from the NOISEX-91 database [24] with varying loudness level of -5db, 0db, 5db, 10db, 15db, and 20db. Each record lasts 4 seconds at 16 kHz sampling rate. NTCD-TIMIT consists of 5 hours of training data, 1 hour of validation data and 1 hour of testing data.

LRS3 +WHAM!: We follow Martel et al. [16] to generate 50,000 pairs of audio records from LRS3 [1] as the training set, 5000 pairs of audio records as the validation set, and 3000 pairs of audio records as the testing set. Every LRS3 record is cut to 2-second long clip at 16 kHz sampling rate then mixed with sample noise record from WHAM! [25] dataset.

B. Evaluation

We use Scale-invariant signal-to-distortion ratio improvement (SI-SDRi) metric as the primary criterion to evaluate the output speech because it reflects the noise reduction and speech improvement brought by the separation. For completeness, we also report two other metrics: a) Perceptual Evaluation of Speech Quality (PESQ) [22] and b) Extended Short-time Objective Intelligibility (ESTOI) [8].

C. Performance

Table I reports the evaluation results on NTCD-TIMIT and LRS3+WHAM!, comparing the audio separation qualities on the testing split of the two datasets using the benchmark models and our proposed BIN model. We also investigate the complexity and compute cost of the models in Table II. These experiments are run on NVIDIA A100 GPU on a sample input of batch size 1 from LRS3+WHAM!.

We observe that for both datasets, BIN with different number of iterations brings performance improvement to all the three evaluation metrics compared with the lightweight AVLIT and IIA-Net performance from our replication study. Our most lightweight model, BIN/8 iterations, has computational complexity comparable to AVLIT, while yielding slight performance improvement on LRS3+WHAM! and significant improvement on NTCD-TIMIT.

[1]The replicated result of AVLIT in our experiment is comparable but different from what is reported in the original paper, because the noise distribution is different from that in Martel et al. [16]. This is because there are no standard settings of the noise. We follow the same noise dB range as described in Martel et al. [16] but noise sample using our own scripts, therefore yielding different noise sample and noise dB level for every audio mixture sample.

Compared to RTFS-Net, our BIN/16 iterations and BIN/12 iterations have better performance on the main separation quality estimation metric SI-SDRi on NTCD-TIMIT and LRS3+WHAM! respectively, while using only 55% and 26% of RTFS-Net training time. Table II shows that these models also cut the GPU inference time by 82% and 74% respectively.

D. Ablation Study

We investigate various modifications of the reported BIN/12 iterations with respect to the fusion mechanism on LRS3+WHAM! and report results in Table III. The first row report the model performance with one single generator function $\mathcal{G}$, instead of two distinct feature generator functions $\mathcal{G}_A$ and $\mathcal{G}_V$ for audio and video. In this case, the input to the single generator function is the fused input of audio and video embeddings, and no fusion token is needed. We can observe a performance decrease compared to BIN, showing that having separate fusion tokens is helpful. However we also note that even the simplified model structure outperforms other lightweight models on noisy AVSS.

Next, we ablate the different fusion tokens to evaluate their impact on the performance. BIN-No $\mathbf{c}^A$ removes the audio fusion token and only allows the audio modality to receive video fusion tokens. Similarly, only the video modality can access the audio fusion tokens in BIN-No $\mathbf{c}^V$. Due to the missing information from the other modality, performance is lower in both experiments. In addition, BIN-No $\mathbf{c}$ removes the variable fusion function $\mathcal{F}$ and $\mathbf{c}_i$, only allowing the two tokens to iterate through their own modality pipelines, which is similar to a late fusion design.

E. Further Analysis

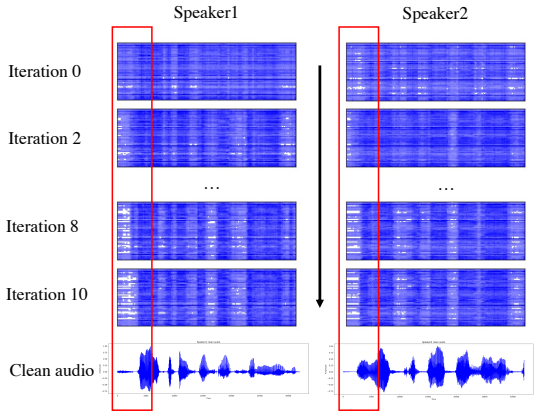

Fig. 2: The progression of the latent masks across fusion iterations of the final trained BIN model on one LRS3+WHAM! test sample. The masks in the later iterations show patterns more aligned with the actual clean audio.

We next study how BIN refines the task across multiple fusion iterations. Using the BIN/12 iterations model trained for LRS3+WHAM!, we compute the audio separation masks $\mathbf{m}^{(i)}$ at each fusion iteration by applying the predictive function $\mathcal{P}$ to the intermediate audio and video features generated at

Dataset	Model	**SI-SDRi** ↑	PESQ ↑	ESTOI ↑	Training time (h)↓
NTCD-TIMIT	AVConvTasNet[26] *	9.02	1.33	0.40	-
	LAVSE[2] *	6.22	1.31	0.37	-
	L2L[3] *	3.36	1.23	0.26	-
	IIA-Net [13] **	6.04	1.33	0.38	2.92
	AVLIT [16] [1]	8.59	1.39	0.44	3.88
	RTFS-Net [21] **	11.28	1.78	0.58	23.17
	BIN/8 Iterations	10.68	1.51	0.50	5.45
	BIN/12 Iterations	10.87	1.51	0.51	7.92
	BIN/16 Iterations	**11.62**	1.57	0.53	12.71
LRS3+WHAM!	AVConvTasNet[26] *	6.21	1.29	0.60	-
	LAVSE[2] *	5.59	1.24	0.50	-
	L2L [3] *	7.60	1.16	0.51	-
	IIA-Net [13] **	8.93	1.36	0.55	16.74
	AVLIT [16]	11.62	1.53	0.65	28.09
	RTFS-Net [21] **	12.14	1.74	0.70	193.45
	BIN/8 Iterations	11.82	1.55	0.66	34.55
	BIN/12 Iterations	**12.25**	1.59	0.68	50.58
	BIN/16 Iterations	10.84	1.49	0.53	81.91

TABLE I: Test performance on NTCD-TIMIT and LRS3+WHAM!. * results are as reported in earlier literature. ** these models were originally designed for speech extraction task and then adapted by us to speech separation task.

Model	MACs (G)	#Params (M)	GPU Inference Time (s)
IIA-Net	9.93	3.52	0.03
AVLIT	36.76	5.72	0.02
RTFS-Net	67.38	1.07	0.23
BIN/8 iterations	42.95	6.05	0.03
BIN/12 iterations	63.58	6.05	0.04
BIN/16 iterations	84.22	6.05	0.06

TABLE II: Comparison with baseline models in terms of computational efficiency with a sample of batch size 1 from LRS3+WHAM!.

Model	SI-SDRi	PESQ	ESTOI
BIN-Single $\mathcal{G}$	11.99	1.57	0.67
BIN-No $\mathbf{c}^A$	11.53	1.53	0.65
BIN-No $\mathbf{c}^V$	11.55	1.52	0.65
BIN-No $\mathbf{c}$	9.64	1.43	0.60
BIN (Full)	12.25	1.59	0.68

TABLE III: Ablation study of BIN/12 iterations on LRS3+WHAM!.

each fusion iteration $\hat{\mathbf{a}}^{(i)}$ and $\hat{\mathbf{v}}^{(i)}$. In Fig 2, we visualize the mask $\mathbf{m}$ as generated after each fusion along with the clean signal for a sample mixture. It clearly shows the peaks and troughs change and become sharper in estimated masks of later iterations, matching the patterns in the clean speaker-specific audio. This qualitatively indicates that the model is progressively refining the mask through the fusion iterations.

Quantitatively, we demonstrate the overall progression of audio separation of the same model across fusion iterations on the test-set of LRS3+WHAM! in Fig 3. We observe an initial sharp and then steady increase in output quality throughout the twelve iterations in the BIN/12 iterations model.

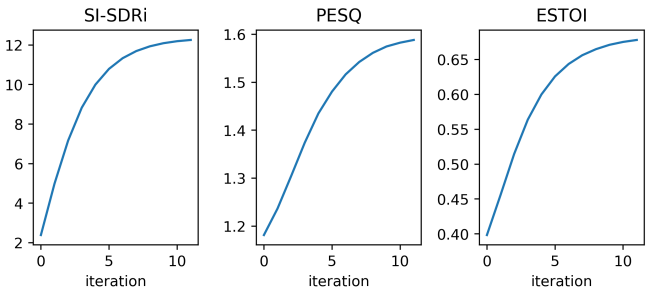

Fig. 3: The progression of the audio separation quality through BIN iterations in the trained BIN model on LRS3+WHAM! testing split. The iterations are not epochs of training, but the unrolling iterations R of the model during test time.

IV. Conclusion

Our paper presents a new approach for noisy AVSS that incorporates backward connections to reduce compute complexity, while retaining near SOTA performance. In our experiments on two benchmark datasets designed for the noisy AVSS task, we show that BIN can match and even slightly improve SOTA models performance on AVSS while cutting the GPU training time and GPU inference time significantly.

References

[1] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. 2018. LRS3-TED: a large-scale dataset for visual speech recognition. arXiv:1809.00496 [cs.CV]

[2] Shang-Yi Chuang, Yu Tsao, Chen-Chou Lo, and Hsin-Min Wang. 2020. Lite Audio-Visual Speech Enhancement. arXiv:2005.11769

[3] Ariel Ephrat, Inbar Mosseri, Oran Lang, Tali Dekel, Kevin Wilson, Avinatan Hassidim, William T. Freeman, and Michael Rubinstein. 2018. Looking to listen at the cocktail party: a speaker-independent audio-visual model for speech separation. *ACM Transactions on Graphics* 37, 4 (July 2018), 1–11.

[4] Ruohan Gao and Kristen Grauman. 2021. VisualVoice: Audio-Visual Speech Separation with Cross-Modal Consistency. arXiv:2101.03149 [cs.CV]

[5] Naomi Harte and Eoin Gillen. 2015. TCD-TIMIT: An Audio-Visual Corpus of Continuous Speech. *IEEE Transactions on Multimedia* 17, 5 (2015), 603–615. https://doi.org/10.1109/TMM.2015.2407694

[6] Richard C Hendriks, Timo Gerkmann, and Jesper Jensen. 2013. *DFT-domain based single-microphone noise reduction for speech enhancement*. Springer.

[7] Xiaolin Hu, Kai Li, Weiyi Zhang, Yi Luo, Jean-Marie Lemercier, and Timo Gerkmann. 2021. Speech Separation Using an Asynchronous Fully Recurrent Convolutional Neural Network. In *NeurIPS*.

[8] Jesper Jensen and Cees H. Taal. 2016. An Algorithm for Predicting the Intelligibility of Speech Masked by Modulated Noise Maskers. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 24, 11 (2016), 2009–2022.

[9] Fahad Shahbaz Khan, Rao Muhammad Anwer, Joost Van De Weijer, Andrew D Bagdanov, Maria Vanrell, and Antonio M Lopez. 2012. Color attributes for object detection. In *2012 IEEE CVPR*. IEEE, 3306–3313.

[10] Jiyoung Lee, Soo-Whan Chung, Sunok Kim, Hong-Goo Kang, and Kwanghoon Sohn. 2021. Looking into your speech: Learning cross-modal affinity for audio-visual speech separation. In *Proceedings of CVPR*. 1336–1345.

[11] Chenda Li and Yanmin Qian. 2020. Deep audio-visual speech separation with attention mechanism. In *ICASSP 2020*. IEEE, 7314–7318.

[12] Kai Li, Fenghua Xie, Hang Chen, Kexin Yuan, and Xiaolin Hu. 2024. An Audio-Visual Speech Separation Model Inspired by Cortico-Thalamo-Cortical Circuits. arXiv:2212.10744 [cs.SD]

[13] Kai Li, Runxuan Yang, Fuchun Sun, and Xiaolin Hu. 2024. IIANet: An Intra- and Inter-Modality Attention Network for Audio-Visual Speech Separation. arXiv:2308.08143

[14] Jiuxin Lin, Xinyu Cai, Heinrich Dinkel, Jun Chen, Zhiyong Yan, Yongqing Wang, Junbo Zhang, Zhiyong Wu, Yujun Wang, and Helen Meng. 2023. Av-sepformer: Cross-attention sepformer for audio-visual target speaker extraction. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1–5.

[15] Yi Luo and Nima Mesgarani. 2019. Conv-TasNet: Surpassing Ideal Time–Frequency Magnitude Masking for Speech Separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 27, 8 (2019), 1256–1266.

[16] Héctor Martel, Julius Richter, Kai Li, Xiaolin Hu, and Timo Gerkmann. 2023. Audio-Visual Speech Separation in Noisy Environments with a Lightweight Iterative Model. arXiv:2306.00160 [eess.AS]

[17] Harry McGurk and John MacDonald. 1976. Hearing lips and seeing voices. *Nature* 264, 5588 (1976), 746–748.

[18] Daniel Michelsanti, Zheng-Hua Tan, Shi-Xiong Zhang, Yong Xu, Meng Yu, Dong Yu, and Jesper Jensen. 2021. An overview of deep-learning-based audio-visual speech enhancement and separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 29 (2021), 1368–1396.

[19] Arsha Nagrani, Shan Yang, Anurag Arnab, Aren Jansen, Cordelia Schmid, and Chen Sun. 2022. Attention Bottlenecks for Multimodal Fusion. (2022). arXiv:2107.00135 [cs.CV]

[20] Sarah Partan and Peter Marler. 1999. Communication goes multimodal. *Science* 283, 5406 (1999), 1272–1273.

[21] Samuel Pegg, Kai Li, and Xiaolin Hu. 2024. RTFS-Net: Recurrent Time-Frequency Modelling for Efficient Audio-Visual Speech Separation. arXiv:2309.17189

[22] A.W. Rix, J.G. Beerends, M.P. Hollier, and A.P. Hekstra. 2001. Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs. In *2001 IEEE ICASSP*, Vol. 2. 749–752 vol.2.

[23] Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, and Ashish Vaswani. 2021. Bottleneck transformers for visual recognition. In *Proceedings of CVPR*. 16519–16529.

[24] Andrew Varga and Herman J.M. Steeneken. 1993. Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems. *Speech Communication* 12, 3 (1993), 247–251.

[25] Gordon Wichern, Joe Antognini, Michael Flynn, Licheng Richard Zhu, Emmett McQuinn, Dwight Crow, Ethan Manilow, and Jonathan Le Roux. 2019. WHAM!: Extending Speech Separation to Noisy Environments. arXiv:1907.01160 [cs.SD]

[26] Jian Wu, Yong Xu, Shi-Xiong Zhang, Lian-Wu Chen, Meng Yu, Lei Xie, and Dong Yu. 2019. Time Domain Audio Visual Speech Separation. arXiv:1904.03760 [eess.AS]