Looking into Your Speech: Learning Cross-modal Affinity for Audio-visual Speech Separation -Supplementary Materials -

Jiyoung Lee[1*] easy00@yonsei.ac.kr
Soo-Whan Chung[1,2*] soowhan.chung@navercorp.com
Sunok Kim[3] sunok.kim@kau.ac.kr
Hong-Goo Kang[1†] hgkang@yonsei.ac.kr
Kwanghoon Sohn[1†] khsohn@yonsei.ac.kr

[1]Department of Electrical & Electronic Engineering, Yonsei University, Korea
[2]Naver Corporation, Korea [3]Korea Aerospace University, Korea

https://caffnet.github.io/

In this document, we describe architecture details of CaffNet and CaffNet-C, details of implementation and training, and provide more quantitative and qualitative results on LRS2, LRS3, and VoxCeleb2 datasets.

1. Network

1.1. Complex-valued Networks

Here we present a brief review of complex-valued networks, which can handle complex computations in deep networks [1, 2, 3]. The complex-valued convolution operation on the intermediate feature representation $\mathbf{h} = \mathbf{h}_r + i\mathbf{h}_i$ with a complex-valued convolutional filter $\mathbf{w} = \mathbf{w}_r + i\mathbf{w}_i$:

$$\mathbf{w} * \mathbf{h} = (\mathbf{h}_r * \mathbf{x}_r - \mathbf{h}_i * \mathbf{w}_i) + i(\mathbf{h}_r * \mathbf{w}_i + \mathbf{h}_i * \mathbf{w}_r), \quad (1)$$

where $\mathbf{w}_r, \mathbf{w}_i$ are real-valued matrices of filter and $\mathbf{h}_r, \mathbf{h}_r$ are real-valued matrices of complex feature representation. In practice, complex convolutions can be implemented as two different real-valued convolution operations with shared real-valued convolution filters as follows [1], and activation functions like ReLU were also adapted to the complex domain. To establish the complex-valued convolutional layer in our networks, we modified 2D complex-valued convolutional layers [1] into 1D complex-valued convolutional layers.

1.2. Network Architecture

Our networks are based on V-Conv [4] architecture, which consists of encoder-decoder architecture. For the audio-visual encoder, it contains two coupled encoders; an audio encoder and a visual encoder. In Tab. 1, CaffNet follows basic configuration of V-Conv [4]. Different from that, the affinity module is added for audio-visual fusion considering global and local correspondence between cross-modal streams. As demonstrated in Tab. 2, CaffNet-C leverages complex-valued convolutional layers ($\mathbb{C}$onv) in audio encoder and mask decoder. For the simplicity, we denote $\Pi(\mathcal{E}_f(I^{1:5}), \cdots, \mathcal{E}_f(I^{T\text{-}4:T}))$ as $\mathcal{E}_f(I)$ in Tab. 1 and Tab. 2.

2. Implementation and Training Details

Visual Feature Extraction. Visual features for audio-visual speech separation represent implicit information correspondent to target speech, *i.e*. linguistic representation on lip movements, called *viseme*. We train the feature extractor using the strategy of audio-to-video synchronization in self-supervision. In [5, 6], they build two-stream networks to embed audio and visual sequences onto a latent space. In their strategies, when input audio and visual segments are taken from the same timestamps, the distance between audio-visual embeddings is minimized, whereas it is maximized for the segments from different offsets. As a result, the embedding learns linguistic information commonly existing on speech sound and lip movements. Especially, we adopt M-way matching method proposed in [6, 7] for the representation learning by providing multiple negative samples over a single positive input,

$$E = -\frac{1}{2N}\sum_{n=1}^{N}\sum_{m=1}^{M} y_{n,m}\log(p_{n,m})$$
$$p_{n,m} = \frac{\exp(d_{n,m}^{-1})}{\sum_{m=1}^{M}\exp(d_{n,m}^{-1})} \quad (2)$$

where $d_{n,m} = ||v_{n,m} - a_{n,m}||_2$ is similarity distance between embedding pairs and $y_{n,m} \in \{0, 1\}$ denotes simi-

Module	Layer	K.	Ch. I/O	S.	P.	B.N.	Act.	Input	Output
Visual Encoder	V-Conv1D	5	512/1536	1	2	✓	ReLU	$\mathcal{E}_f(I)$	v-feat1
	V-Conv1D ($\times$8)	5	1536/1536	1	2	✓	ReLU	v-feat1	v-feat9
	V-Conv1D	5	1536/1536	1	2	-	-	v-feat9	$\mathbf{V}$
Audio Encoder	Decompose	-	257/257	-	-	-	-	$\mathbf{X}$	$\lvert\mathbf{X}\rvert$
	A-Conv1D	5	257/1536	1	2	✓	ReLU	$\lvert\mathbf{X}\rvert$	a-feat1
	A-Conv1D ($\times$3)	5	1536/1536	1	2	✓	ReLU	a-feat1	a-feat4
	A-Conv1D	5	1536/1536	1	2	-	-	a-feat4	$\mathbf{S}$
Affinity Module	V-Nonlocal ($\times$2)	5	1536/1536	1	2	✓	ReLU	$\mathbf{V}$	$\bar{\mathbf{V}}$
	A-Nonlocal ($\times$2)	5	1536/1536	1	2	✓	ReLU	$\mathbf{S}$	$\bar{\mathbf{S}}$
	AV-Aff.	-	1536/1536	-	-	-	-	$\bar{\mathbf{S}}$, $\bar{\mathbf{V}}$	$\hat{\mathbf{V}}$
	Concat. (Π)	-	1536/3072	-	-	-	-	$\bar{\mathbf{S}}$, $\hat{\mathbf{V}}$	$\mathbf{\Psi}$
Mask Decoder	Conv1D	5	3072/1536	1	2	✓	ReLU	$\mathbf{\Psi}$	m-feat1
	Conv1D ($\times$13)	5	1536/1536	1	2	✓	ReLU	m-feat1	m-feat14
	Conv1D	5	1536/257	1	2	✓	Sigmoid	m-feat14	$\mathbf{M}$
	Mult. ($\odot$)	-	-	-	-	-	-	$\lvert\mathbf{X}\rvert$, $\mathbf{M}$	$\hat{\mathbf{Y}}$

Table 1: Network configuration of CaffNet. 'K', 'S' and 'P' represents the kernel, stride, and padding size of convolution layer, and 'Ch. I/O' represents channels of input and output relative to the input, respectively. 'B.N' is the batch normalization layer and 'Act.' is the activation function.

Module	Layer	K.	Ch. I/O	S.	P.	B.N.	Act.	Input	Output
Visual Encoder	V-Conv1D	5	512/1536	1	2	✓	ReLU	$\mathcal{E}_f(I)$	v-feat1
	V-Conv1D ($\times$8)	5	1536/1536	1	2	✓	ReLU	v-feat1	v-feat9
	V-Conv1D	5	1536/1536	1	2	-	-	v-feat9	$\mathbf{V}$
Audio Encoder	A-$\mathbb{C}$Conv1D	5	257/1536	1	2	✓	LReLU	$\mathbf{X}$	a-feat1
	A-$\mathbb{C}$Conv1D ($\times$3)	5	1536/1536	1	2	✓	LReLU	a-feat1	a-feat4
	A-$\mathbb{C}$Conv1D	5	1536/1536	1	2	-	-	a-feat4	$\mathbf{S}$
	Decompose	-	-	-	-	-	-	$\mathbf{S}$	$\lvert\mathbf{S}\rvert$, $e^{i\theta_\mathbf{S}}$
Affinity Module	V-Nonlocal ($\times$2)	5	1536/1536	1	2	✓	ReLU	$\mathbf{V}$	$\bar{\mathbf{V}}$
	A-Nonlocal ($\times$2)	5	1536/1536	1	2	✓	ReLU	$\lvert\mathbf{S}\rvert$	$\lvert\bar{\mathbf{S}}\rvert$
	AV-Aff.	-	1536/1536	-	-	-	-	$\lvert\bar{\mathbf{S}}\rvert$, $\bar{\mathbf{V}}$	$\hat{\mathbf{V}}$
	Concat. (Π)	-	1536/3072	-	-	-	-	$\lvert\bar{\mathbf{S}}\rvert$, $\hat{\mathbf{V}}$	$\lvert\mathbf{\Psi}\rvert$
	Reconstruct	-	-	-	-	-	-	$\lvert\mathbf{\Psi}\rvert$, $e^{i\theta_\mathbf{S}}$	$\mathbf{\Psi}$
Mask Decoder	$\mathbb{C}$Conv1D	5	3072/1536	1	2	✓	LReLU	$\mathbf{\Psi}$	m-feat1
	$\mathbb{C}$Conv1D ($\times$13)	5	1536/1536	1	2	✓	LReLU	m-feat1	m-feat14
	$\mathbb{C}$Conv1D	5	1536/257	1	2	-	Tanh	m-feat14	$\mathbf{M}$
	Mult. ($\odot$)	-	-	-	-	-	-	$\mathbf{X}$, $\mathbf{M}$	$\hat{\mathbf{Y}}$

Table 2: Network configuration of CaffNet-C. 'K', 'S' and 'P' represents the kernel, stride, and padding size of convolution layer, and 'Ch. I/O' represents channels of input and output relative to the input, respectively. 'B.N' is the batch normalization layer and 'Act.' is the activation function. 'LReLU' indicates LeakyReLU function with a slope 0.2. $\mathbb{C}$onv denotes complex-valued convolutional layer. At the last layer of mask decoder, we adopt tanh activation for the magnitude of the estimated mask.

larity label. In this experiment, we stacked 5 consecutive visual frames of 224×224 pixels with RGB channels as the visual input, and 20 audio frames as the audio input, where they are 0.2-seconds length. The architecture configuration is same as described in [6]. It shows powerful performance compared to using contrastive loss, and we prepare two visual feature extractors; one is for LRS2 and LRS3 dataset and the other is for VoxCeleb2 dataset. We pre-trained visual feature extractors and not fine-tuned on the separation task; thus there is still a margin to be improved with joint training of the extractor and the separator.

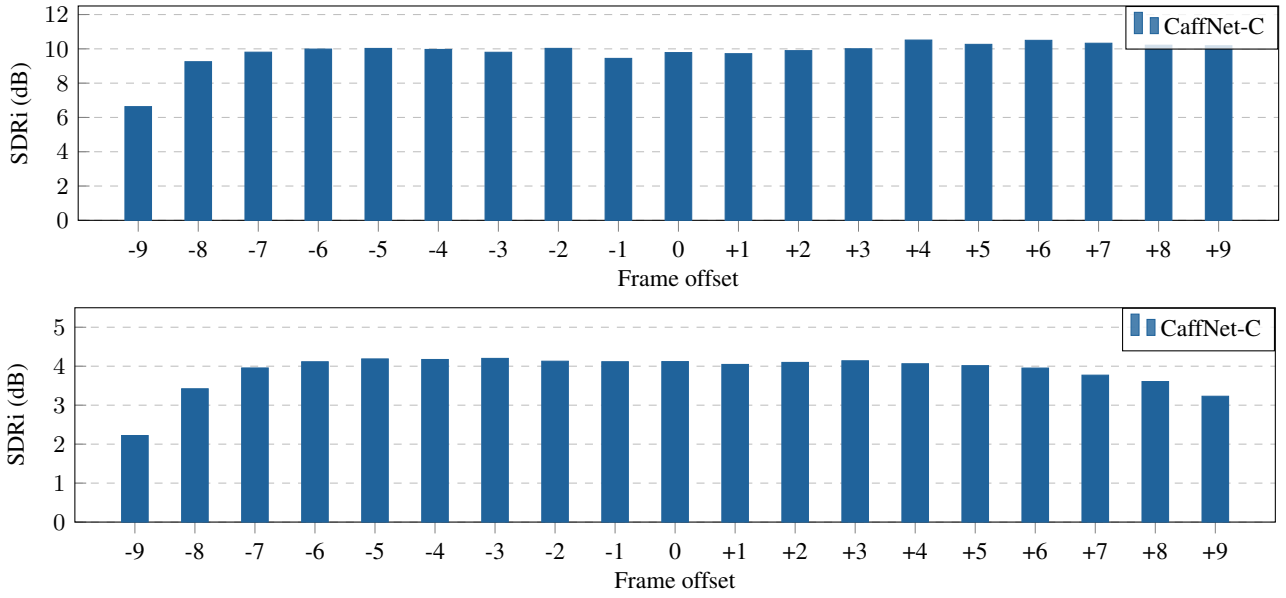


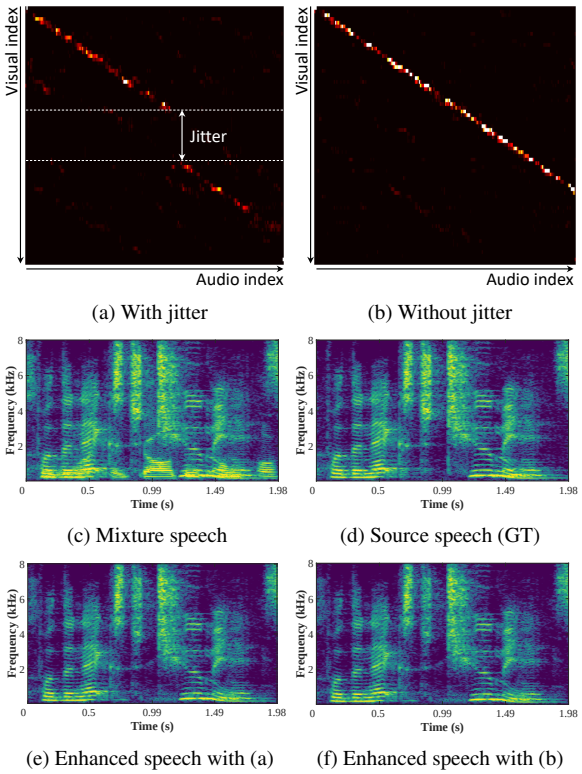
Figure 1: Speech separation performance with respect to each delay offset between audio and visual streams on LRS3 (top) and VoxCeleb2 (bottom) datasets. The frame offset unit is 40ms which is the duration length between consecutive video frames.

Training and Optimization We use PyTorch library [8] and single RTX 24GB to train our networks. The network parameters are optimized using the mini-batch stochastic gradient descent (SGD) method. All networks are trained from scratch and we applied the Adam optimizer [9] with learning rate 1×10^{-5}, beta1 $\beta_1 = 0.9$, beta2 $\beta_2 = 0.999$ and batch size 32. We use PyTorch's ReduceLROnPlateau learning rate scheduler with a reduction factor of 0.8 and a patience parameter of 2 to adapt the learning rate during training. At each training iteration, we randomly sampled one audio-video pair and another audio that has a different identity. We randomly take segments from each audio signal and add them to make mixture input. Hence, we use 64K training samples for each epoch and all networks are trained for a total of 50 epochs.

3. More Results

3.1. Quantitative Results

We provide more quantitative results for AVSS performance regarding SDR improvement (SDRi) metric concerning varying delay in Fig. 1 on LRS3 and VoxCeleb2 datasets. The results are obtained from CaffNet-C using predicted magnitude and phase. On the top of Fig. 1, we report the average SDRi of unseen 1000 speaker samples on LRS3 dataset. In this experiment, we train CaffNet-C using LRS2 dataset only, so the network cannot see any samples of LRS3 at the training stage. At the bottom of Fig. 1, we provide the average SDRi of unseen 1000 speaker samples on VoxCeleb2 dataset.

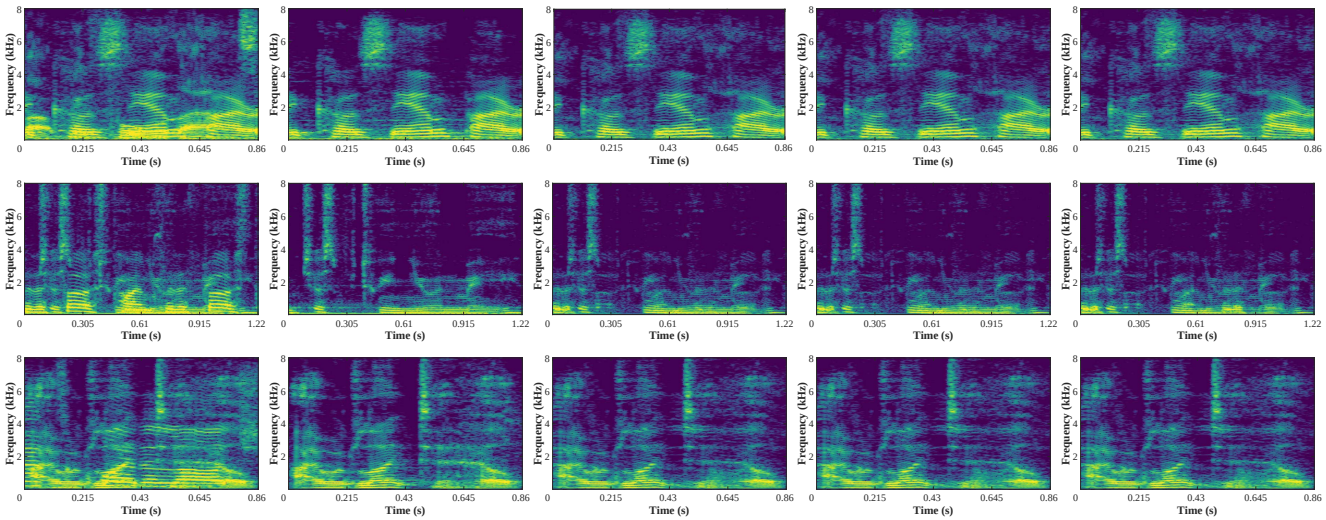


(a) With jitter (b) Without jitter

(c) Mixture speech (d) Source speech (GT)

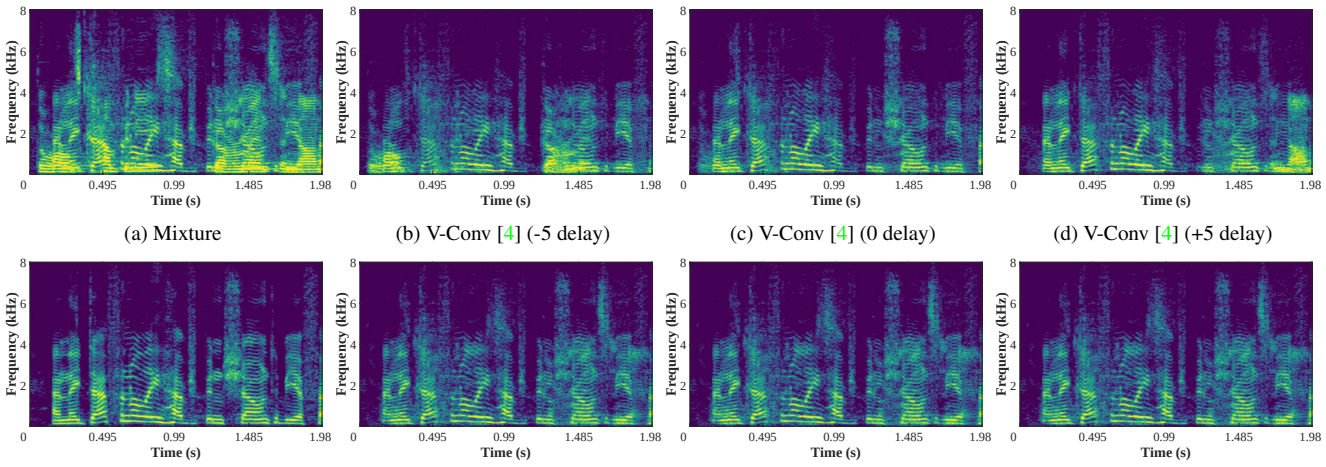
(e) Enhanced speech with (a) (f) Enhanced speech with (b)

Figure 2: Qualitative evaluation of affinity matrices and spectrograms with respect to jitter effect on LRS2 dataset. The results are obtained from CaffNet-C.



(a) Mixture (b) Source (GT) (c) -5 delay (d) 0 delay (e) +5 delay

Figure 3: Qualitative evaluation of spectrograms with respect to the frame delay on LRS2 dataset. (c), (d) and (e) are results of CaffNet-C. Each row is different testing sample and all results are reconstructed from the predicted magnitude and phase.



(a) Mixture (b) V-Conv [4] (-5 delay) (c) V-Conv [4] (0 delay) (d) V-Conv [4] (+5 delay)

(e) Source (GT) (f) CaffNet (-5 delay) (g) CaffNet (0 delay) (h) CaffNet (+5 delay)

Figure 4: Qualitative comparison results of spectrogram estimated by V-Conv [4] and CaffNet on LRS3 dataset. All results are reconstructed from the combination of estimated magnitude and ground-truth phase.

3.2. Qualitative Results of Spectrograms

We provide more qualitative results of spectrograms. Fig. 2 shows the examples with and without jitter. Fig. 3 includes the spectrogram results of three different testing samples. In this experiment, we sample by immobilizing the audio stream at a certain time and moving the video segment according to the degree of delay. Although the video stream does not synchronize exactly with the audio stream, the results are similarly predicted. In Fig. 4, we compare the results with V-Conv [4]. These results validate that the CaffNet are flexible and interpretable for the delay, thus widely applicable on real-world videos. Fig. 5 demonstrates the results of spectrograms on VoxCeleb2 dataset. At the last column in Fig. 5, colored image means the target speaker and gray-scale image indicates the interfering speaker. The target speaker's speech is isolated while the other's interfering speech is suppressed.

3.3. Video

Our demo video[1] includes the examples of videos on LRS2 dataset, and the examples from real-world news videos.

[1] https://youtu.be/9R2qQ7dGTp8

(a) Mixture speech (b) Target speech (c) Enhanced speech

Figure 5: Examples of speech separation results on VoxCeleb2 dataset. From left to right, each column notes mixed speech, target speech and separated output. These are randomly selected samples of CaffNet-C with full prediction of magnitude and phase components.

References

[1] Hyeong-Seok Choi, Jang-Hyun Kim, Jaesung Huh, Adrian Kim, Jung-Woo Ha, and Kyogu Lee. Phase-aware speech enhancement with deep complex u-net. *arXiv preprint arXiv:1903.03107*, 2019.

[2] Yuan-Shan Lee, Chien-Yao Wang, Shu-Fan Wang, Jia-Ching Wang, and Chung-Hsien Wu. Fully complex deep neural network for phase-incorporating monaural source separation. *In: ICASSP*, 2017.

[3] Chiheb Trabelsi, Olexa Bilaniuk, Ying Zhang, Dmitriy Serdyuk, Sandeep Subramanian, João Felipe Santos, Soroush Mehri, Negar Rostamzadeh, Yoshua Bengio, and Christopher J Pal. Deep complex networks. *In: ICLR*, 2018.

[4] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. The conversation: Deep audio-visual speech enhancement. *In: INTERSPEECH*, 2018.

[5] Joon Son Chung and Andrew Zisserman. Lip reading in the wild. *In: ACCV*, 2016.

[6] Soo-Whan Chung, Joon Son Chung, and Hong-Goo Kang. Perfect match: Improved cross-modal embeddings for audio-visual synchronisation. *In: ICASSP*, 2019.

[7] Soo-Whan Chung, Joon Son Chung, and Hong-Goo Kang. Perfect match: Self-supervised embeddings for cross-modal retrieval. *IEEE Journal of Selected Topics in Signal Processing*, 14(3):568–576, 2020.

[8] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.