Supplementary Material for Speaker-adaptive Lip Reading with User-dependent Padding

1 LRW-ID data split information

Table 1 shows the number of speakers, the number of word classes, and the total number of videos for each train, adaptation (validation), and test splits of LRW-ID dataset.

2 Comparison with finetuning methods

In order to validate the effectiveness of the proposed user-dependent padding, we compare the adaptation performances with different finetuning methods. To this end, three different parts of network are employed for finetuning as follows, 1) finetuning only the classifier (*i.e.*, the last fully connected layer) (denoted Finetune_classifier), 2) finetuning the last TCN block and the classifier (denoted Finetune_partial), and 3) finetuning the entire network (denoted Finetune_whole). The comparison results on LRW-ID are shown in Fig. 1. The number of network parameters for the 20 speakers are 47.8M, 171.3M, and 811.6M for Finetune_classifier, Finetune_partial, and Finetune_whole, respectively. On the other hand, the proposed user-dependent padding requires 43.58M parameters. By examining the adaptation results, we can confirm that when a small number of adaptation data is accessible ($\sim$50%), the proposed user-dependent padding is the most effective in speaker adaptation. When the adaptation data is sufficient, the method Finetune_whole achieves the best performance as one can expect. We found that finetuning the last classifier only is not effective for any ratio of adaptation data.

3 Network Architecture

3.1 User-dependent Padding

The network architecture for sentence-level lip reading (GRID) is shown in Table 1. It has a modified architecture of LipNet [2]. The user-dependent padding is inserted instead of the zero-padding used during pre-training, for all padded convolutions in front-end. For the 3D convolution, only spatial padding is changed with the user-dependent padding while that of the temporal dimension remains to zero since the temporal length can be varying to the inputs in practice. The value of user-dependent padding in the table represents temporal T_p, height H_p, and width W_p size of the padding multiplicated with the channel C_p size

Table 1. Train, Adaptation (Validation), and Test splits of LRW-ID

	Splits		
Info.	**Train**	**Adapt (Val)**	**Test**
# Speaker	17,560	20	20
# Tot. class	500	500	500
# Tot. video	480,378	29,918	29,923

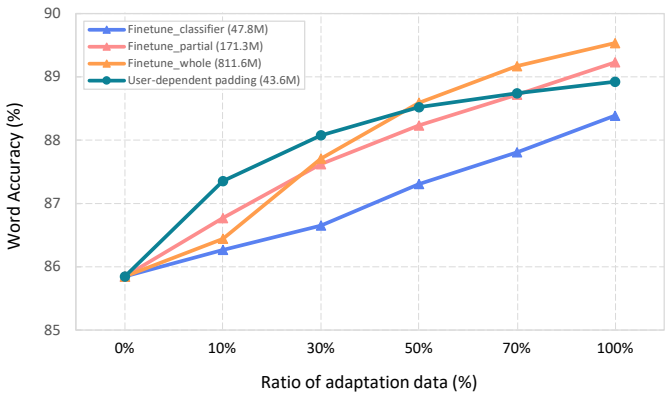

Fig. 1. Comparison results with different finetuning methods on LRW-ID.

(*i.e.*, $[T_p, H_p, W_p] \times C_p$). Moreover, the network architecture based on [4] for the word-level lip reading (LRW-ID) is shown in Table 2. The user-dependent padding is initialized with the padding used during pre-training (*i.e.*, zero) and updated with a learning rate of 0.01. For 1 minute adaptation on LRW-ID, the user-dependent padding converges within 200 steps.

3.2 Baseline Models (Speaker-invariant and Speaker code)

The detailed architecture of the methods, Speaker-invariant and Speaker code, used for comparison are illustrated in Fig. 1.

For the speaker-invariant model, we build a model based on the concept of ASR method [5], which trains the model via adversarial learning to suppress the speaker information from the encoded features. Specifically, an additional speaker classifier is introduced which classifies the speaker identity from the encoded visual feature. The sign of gradient calculated from the speaker classifier is reversed before backpropagated through the front-end [3] using Gradient Reversal Layer (GRL), thus the front-end learns to suppress the speaker information from the encoded visual feature, while the speaker classifier attempts to find the speaker information from the encoded visual feature in an adversarial manner.

For the speaker-adaptive model, we bring a popular speaker-adaptation method [1] of ASR which utilizes speaker code as additional inputs with additional layers. For GRID, we use 128, 64, and 32 dimensions of speaker code with three additional fully connected layers ($W_1 \in \mathbb{R}^{256\times128}$, $W_2 \in \mathbb{R}^{192\times128}$, $W_3 \in \mathbb{R}^{160\times128}$)

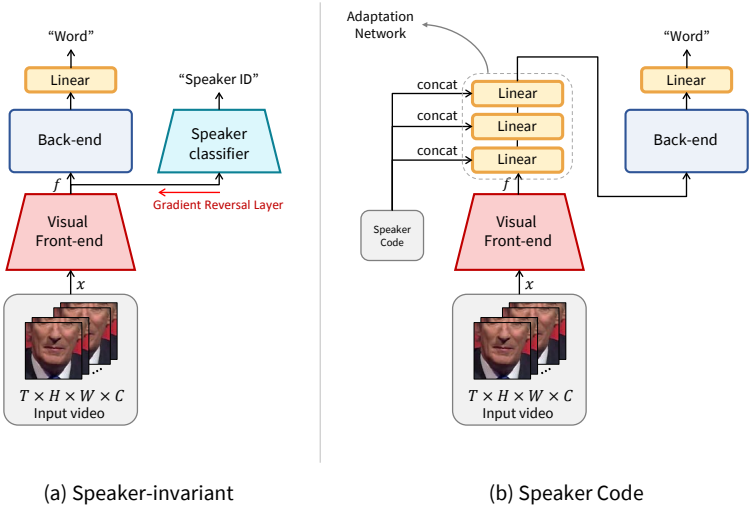

Fig. 2. Illustration of the baseline models. (a) Speaker-invariant speech recognition model trained via adversarial learning. The visual front-end is trained to suppress the speaker information from the encoded visual features. (b) Speaker-adaptive speech recognition model using speaker code. Speaker code is applied to the extracted visual feature with an additional Adaptation Network.

which correspond to Adaptation Network of [1] to transform the visual feature encoded from the front-end. For LRW, 256, 128, and 64 dimensions of speaker code with three additional fully connected layers ($W_1 \in \mathbb{R}^{768\times512}$, $W_2 \in \mathbb{R}^{640\times512}$, $W_3 \in \mathbb{R}^{576\times512}$) are utilized. The training procedures are as follows, 1) bring a pre-trained lip reading model, 2) only train Adaptation Network and the speaker code after attaching them to the pre-trained model using the training dataset $\mathcal{S}$ while other network parameters are fixed, and 3) perform adaptation by training speaker code only using the adaptation dataset $\mathcal{A}$.

Please note that different from the proposed method, the speaker-invariant model and the speaker code model require the speaker labels for the whole training dataset $\mathcal{S}$ which is usually composed of very large speakers and utterances. In contrast, the proposed method only requires the speaker label for the adaptation data instead of the training data.

4 Adaptation results of each speaker

The adaptation results of each speaker by using the different rate of adaptation data are shown in Table 3 and Table 4, for GRID and LRW-ID, respectively. By increasing the amount of adaptation data, we can generally increase the lip reading performance of each speaker. Moreover, the unsupervised adaptation results of each speaker can be found in Table 5 and Table 6, for GRID and LRW-ID,

Table 2. Sentence-level lip reading architecture.

Network architecture for GRID: input size 75 × 64 × 128 × 3 (T × H × W × C)			
Layer	**Filter size / number / stride**	**User-dependent Padding**	**Output dimensions**
Conv 3D	3 × 5 × 5 / 32 / [1, 2, 2]	[0, 2, 2] × 3	75 × 32 × 64 × 32
Maxpool 2D	2 × 2 / - / [2, 2]	-	75 × 16 × 32 × 32
Conv 3D	3 × 5 × 5 / 64 / [1, 1, 1]	[0, 2, 2] × 32	75 × 8 × 16 × 64
Maxpool 2D	2 × 2 / - / [2, 2]	-	75 × 4 × 8 × 64
Conv 3D	3 × 3 × 3 / 96 / [1, 1, 1]	[0, 1, 1] × 64	75 × 4 × 8 × 96
Maxpool 2D	2 × 2 / - / [2, 2]	-	75 × 2 × 4 × 96
Conv 2D	3 × 3 / 32 / [2, 2]	[1, 1] × 96	75 × 2 × 4 × 32
Conv 2D	3 × 3 / 64 / [2, 2]	[1, 1] × 32	75 × 1 × 2 × 64
Flatten	-	-	75 × 128
Bi-GRU	256	-	75 × 512
Bi-GRU	256	-	75 × 512
Linear	512 × Num_class	-	75 × Num_class

Table 3. Word-level lip reading architecture.

Network architecture for LRW: input size 29 × 112 × 112 × 1 (T × H × W × C)			
Layer	**Filter size / number / stride**	**User-dependent Padding**	**Output dimensions**
Conv 3D	5 × 7 × 7 / 64 / [1, 2, 2]	[0, 3, 3] × 1	29 × 64 × 64 × 64
Max Pool 3D	1 × 3 × 3 / - / [1, 2, 2]	-	29 × 32 × 32 × 64
ResBlock 2D	3 × 3 / 64 / [1, 1] 3 × 3 / 64 / [1, 1]	[1, 1] × 64 [1, 1] × 64	29 × 32 × 32 × 64
ResBlock 2D	3 × 3 / 64 / [1, 1] 3 × 3 / 64 / [1, 1]	[1, 1] × 64 [1, 1] × 64	29 × 32 × 32 × 64
ResBlock 2D	3 × 3 / 128 / [2, 2] 3 × 3 / 128 / [1, 1]	[1, 1] × 64 [1, 1] × 128	29 × 16 × 16 × 128
ResBlock 2D	3 × 3 / 128 / [1, 1] 3 × 3 / 128 / [1, 1]	[1, 1] × 128 [1, 1] × 128	29 × 16 × 16 × 128
ResBlock 2D	3 × 3 / 256 / [2, 2] 3 × 3 / 256 / [1, 1]	[1, 1] × 128 [1, 1] × 256	29 × 8 × 8 × 256
ResBlock 2D	3 × 3 / 256 / [1, 1] 3 × 3 / 256 / [1, 1]	[1, 1] × 256 [1, 1] × 256	29 × 8 × 8 × 256
ResBlock 2D	3 × 3 / 512 / [2, 2] 3 × 3 / 512 / [1, 1]	[1, 1] × 256 [1, 1] × 512	29 × 4 × 4 × 512
ResBlock 2D	3 × 3 / 512 / [1, 1] 3 × 3 / 512 / [1, 1]	[1, 1] × 512 [1, 1] × 512	29 × 4 × 4 × 512
Flatten	-	-	29 × 8192
Linear	8192 × 512	-	29 × 512
MS-TCN	-	-	29 × 768
Temporal Avg Pool	-	-	768
Linear	768 × Num_class	-	Num_class

respectively. Without using the labeled adaptation dataset, we can also adapt to an unseen speaker by applying the self-training methods in an unsupervised manner. Please note that we expect that we can improve the unsupervised adaptation results by applying more advanced learning algorithms such as adversarial learning, uncertainty-aware self-training, and entropy minimization.

Table 4. Adaptation result by using different rate of adaptation data on GRID

Adapt. min	S1	S2	S20	S22	Mean
Baseline	17.04	9.02	10.33	8.13	11.12
10%	9.13	3.87	6.87	4.37	6.05
30%	7.71	2.94	5.97	4.00	5.15
50%	7.44	2.84	5.97	3.83	5.02
70%	7.34	2.81	5.70	3.60	4.86
100%	6.77	2.64	5.77	3.43	4.65

Table 5. Adaptation result by using different rate of adaptation data on LRW-ID

Adapt. min	S1 (#4243)	S2 (#5125)	S3 (#6003)	S4 (#7184)	S5 (#9335)	S6 (#9368)	S7 (#9438)	S8 (#9653)	S9 (#10209)	S10 (#10293)
Baseline	75.93	80.08	84.13	89.36	77.70	84.53	91.12	77.05	88.46	81.33
10%	78.58	82.50	84.22	89.76	82.31	85.68	92.18	80.10	88.46	83.90
30%	79.12	81.83	86.40	89.94	84.21	85.26	93.01	80.90	88.86	84.44
50%	80.00	82.64	87.07	90.49	84.99	85.68	92.97	81.86	88.99	85.39
70%	81.24	82.50	86.48	90.74	85.33	86.32	93.26	82.34	89.26	86.06
100%	81.59	83.45	86.23	91.01	85.23	86.84	93.37	82.50	90.07	86.06

S11 (#10587)	S12 (#11041)	S13 (#11777)	S14 (#11875)	S15 (#11910)	S16 (#13287)	S17 (#13786)	S18 (#15545)	S19 (#15769)	S20 (#17378)	Mean
73.78	86.83	88.07	85.79	72.69	75.95	81.74	87.01	88.25	86.67	85.85
79.02	87.92	88.07	90.86	73.74	77.27	82.10	87.54	89.32	87.84	87.35
80.11	88.69	88.53	91.64	76.89	78.42	81.86	87.74	89.53	88.51	88.08
79.93	89.20	88.99	92.21	78.57	78.42	82.59	88.08	89.74	89.52	88.52
80.11	89.05	88.88	92.57	80.04	78.42	83.43	88.08	89.96	89.46	88.74
82.46	89.20	88.53	92.71	79.20	78.75	83.19	88.37	89.53	89.74	88.92

Table 6. Unsupervised adaptation result of each speaker on GRID

Adapt. min	S1	S2	S20	S22	Mean
Baseline	17.04	9.02	10.33	8.13	11.12
Proposed Method	13.18	4.25	7.43	4.18	7.24

Table 7. Unsupervised adaptation result of each speaker on LRW-ID

Adapt. min	S1 (#4243)	S2 (#5125)	S3 (#6003)	S4 (#7184)	S5 (#9335)	S6 (#9368)	S7 (#9438)	S8 (#9653)	S9 (#10209)	S10 (#10293)
Baseline	75.93	80.08	84.13	89.36	77.70	84.53	91.12	77.05	88.46	81.33
Proposed	78.23	81.97	85.81	89.85	80.76	86.74	92.32	78.81	89.66	83.76

S11 (#10587)	S12 (#11041)	S13 (#11777)	S14 (#11875)	S15 (#11910)	S16 (#13287)	S17 (#13786)	S18 (#15545)	S19 (#15769)	S20 (#17378)	Mean
73.78	86.83	88.07	85.79	72.69	75.95	81.74	87.01	88.25	86.67	85.85
80.65	87.70	88.88	91.57	75.21	77.92	82.59	87.54	90.60	88.01	87.51

References

1. Abdel-Hamid, O., Jiang, H.: Fast speaker adaptation of hybrid nn/hmm model for speech recognition based on discriminative learning of speaker code. In: 2013 IEEE

International Conference on Acoustics, Speech and Signal Processing. pp. 7942–7946. IEEE (2013)
2. Assael, Y.M., Shillingford, B., Whiteson, S., De Freitas, N.: Lipnet: End-to-end sentence-level lipreading. arXiv preprint arXiv:1611.01599 (2016)
3. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: International conference on machine learning. pp. 1180–1189. PMLR (2015)
4. Martinez, B., Ma, P., Petridis, S., Pantic, M.: Lipreading using temporal convolutional networks. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 6319–6323. IEEE (2020)
5. Meng, Z., Li, J., Chen, Z., Zhao, Y., Mazalov, V., Gong, Y., Juang, B.H.: Speaker-invariant training via adversarial learning. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 5969–5973. IEEE (2018)