ADVERSARIAL TRAINING OF END-TO-END SPEECH RECOGNITION USING A CRITICIZING LANGUAGE MODEL

Alexander H. Liu *Hung-yi Lee* *Lin-shan Lee*

College of Electrical Engineering and Computer Science
National Taiwan University
{r07922013, hungyilee, lslee}@ntu.edu.tw

ABSTRACT

In this paper we proposed a novel Adversarial Training (AT) approach for end-to-end speech recognition using a Criticizing Language Model (CLM). In this way the CLM and the automatic speech recognition (ASR) model can challenge and learn from each other iteratively to improve the performance. Since the CLM only takes the text as input, huge quantities of unpaired text data can be utilized in this approach within end-to-end training. Moreover, AT can be applied to any end-to-end ASR model using any deep-learning-based language modeling frameworks, and compatible with any existing end-to-end decoding method. Initial results with an example experimental setup demonstrated the proposed approach is able to gain consistent improvements efficiently from auxiliary text data under different scenarios.

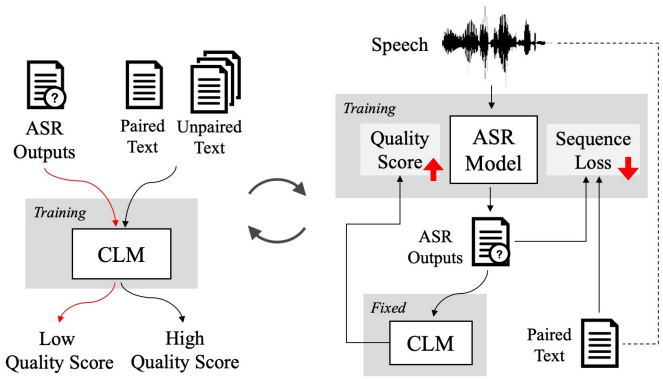
Index Terms— automatic speech recognition, end-to-end, adversarial training, criticizing language model

1. INTRODUCTION

With the fast advances of deep learning technologies, converting the well matured multi-module speech recognition processes [1] into a single speech-to-text model [2] becomes highly attractive. Such end-to-end speech recognition approaches are primarily based on two distinct models: connectionist temporal classification (CTC) [3, 4, 5] and sequence-to-sequence (Seq2seq) [6, 7, 8] models. By introducing an additional blank symbol and a specially defined loss function aggregating many allowed paths within a graph, CTC model can be optimized to generate the correct character sequences from the speech signals regardless of the blank symbols interspersed among. The seq2seq models, on the other hand, simply maximized the likelihood of observing the decoded sequence given the ground truth at every time step. With many recent results [9, 10, 11, 12, 13] approaching the state-of-the-art, end-to-end deep learning has definitely been a very important direction for speech recognition.

Most end-to-end speech recognition approaches require a considerable amount of paired audio-text data, which is costly and time-consuming. Semi-supervised approaches [14, 15, 16, 17] have been developed to address such problem by involving unpaired text data (which are relatively easy to obtain) in the training progress. One approach is to utilize unpaired text data to produce a separately trained language model (LM) to rescore the output of the end-to-end approach [18, 13, 19, 20], but at the price of extra computation during testing. Also, in this way the unpaired text data and paired audio-text data were used separately, and the machines could not learn from them jointly. Another approach is to back-translate (synthesize) speech signals or encoder state sequences [17, 21, 22] from the unpaired text data, so they can be jointly learned in training. However, the improvements achievable with such approaches were limited by the quality of the synthesized data, which is usually far from real.

The Generative Adversarial Networks (GANs) [23] have been shown to be very successful in diversified application areas. Instead of learning from a set of ground truth taken as the upper bound for learning, a generator model and a discriminator model are trained iteratively to challenge and learn from each other step by step. In this paper, inspired by GANs, we propose a novel approach to embed the ad-



(a) CLM learning step (b) ASR model learning step

Fig. 1. Overview of the Adversarial Training (AT) approach for end-to-end speech recognition. The two steps here are conducted iteratively: (a) a Criticizing Language Model (CLM) is trained to evaluate the quality score given a text sequence, and (b) and ASR model is trained to minimize the sequence loss calculated with ground truth while maximizing the scores given by CLM.

vantages of adversarial training (AT) into end-to-end speech recognition. With the proposed approach, huge quantities of unpaired text data can be utilized without a separately trained model, extra computation during testing and the shortcomings of back-translation style data augmentation.

2. PROPOSED APPROACH

2.1. Overview

In our Adversarial Training approach to end-to-end speech recognition, the ASR model is considered as a generator conditioned on the input speech signal whose output is the corresponding transcription. A *Criticizing Language Model* (CLM) is used as a discriminator to distinguish real text from ASR transcriptions. The ASR model and CLM are trained iteratively, so they learn from each other step by step. Fig. 1 gives an overview of the proposed approach.

In Fig. 1(a) for CLM training step, the CLM learns to assign higher scores to real text and lower scores to ASR transcriptions. The real text here does not have to be paired with audio, which is how the unpaired text can be involved in the training processes. This CLM is to evaluate the quality of each given text sequence by offering a score for adversarial purposes, with details given in Sec. 2.2.

In Fig. 1(b) for ASR model learning step, the parameters of CLM are fixed and we train the ASR model by minimizing the sequence loss (e.g. seq2seq loss and/or CTC loss) evaluated with the ground truth just as typical end-to-end training. At the same time, with CLM acting as a discriminator evaluating the quality score for the output of ASR model, the ASR model also has to learn to generate transcriptions obtaining higher quality scores from CLM. The details of the ASR model is in Sec. 2.3.

Note that the ASR model and the CLM are learned iteratively both from scratch. No pre-training is needed. Each of them improves itself based on the challenges offered by the other in each iteration. Once the training ends, the ASR model is expected to implicitly leverage the linguistic knowledge learned from CLM, and the latter is no longer used during testing. This approach can be used with any existing end-to-end speech recognition frameworks and any language modeling framework. Below we take one example set of the proposed approach to explain the details.

2.2. Criticizing Language Model (CLM)

Network Architecture. CLM takes either real text or ASR transcriptions as input and outputs a scalar s as the quality score. The real text is represented as a sequence of one-hot vectors $y = y_1, y_2, ..., y_L$, while for ASR transcriptions this is a sequence of vectors for distributions $\tilde{y} = \tilde{y_1}, \tilde{y_2}, \tilde{y_3}, ...$. Fig. 2 illustrates an example architecture of CLM used in this work. The input vector sequence y (or $\tilde{y}$) is first projected to a lower dimensional space through a single layer neural net. Next, two layers of one-dimensional convolution neural network extracts features for each time index. Finally, average pooling over the time axis is applied to get a single representative feature, which is then transformed to a scalar s (the quality score) with linear projection.

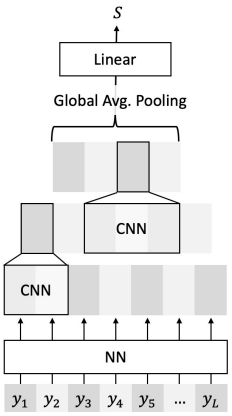


Fig. 2. Network architecture of the CLM.

The reason a convolution-based network instead of a recurrent network is used in Fig 2 is twofold. Convolution with small window size captures local relation features, which can then be averaged over time. Also, CNN based network is relatively more computationally efficient, which is important in adversarial training. But other network architectures such as RNN-LM [13] can also be used here.

Loss Function. A major problem here is that soft distribution vectors produced by the ASR model is very different from one-hot vectors for real text data, making the task of CLM trivial, and the ASR model almost always fail to compete against it. Thanks to Wasserstein GAN (WGAN) [24] which addressed the above problem to some good extent. Based on the concept of WGAN, CLM is designed to estimate the *Earth-Mover* (Wasserstein-1) [25] distance between sequences from real data and ASR output. The loss function of CLM is the weighted sum of a loss L_D and a gradient penalty gp as follows,

$$L_{CLM} = \lambda_{CLM} L_D + \lambda_{gp}\, gp, \tag{1}$$

in which λ_{CLM},λ_{gp} are wieghts and L_D and gp are respectively in Eq (2) and Eq (3) below.

$$L_D = \mathop{\mathbb{E}}_{\tilde{y}\sim\mathbb{P}_a}\big[CLM(\tilde{y})\big] - \mathop{\mathbb{E}}_{y\sim\mathbb{P}_d}\big[CLM(y)\big], \tag{2}$$

where $CLM(y)$ is the quality score for y given by CLM, $\mathbb{P}_a$ the distribution of ASR output $\tilde{y}$ and $\mathbb{P}_d$ the distribution of real text y. $\tilde{y}$ can be sampled from ASR with greedy search and y can be sampled directly from data. The 1-Lipschtiz restriction is imposed for CLM by applying the gradient penalty [26] as below,

$$gp = \mathop{\mathbb{E}}_{\hat{y}\sim\mathbb{P}_{\hat{y}}}\big[(\|\nabla_{\hat{y}} CLM(\hat{y})\| - 1)^2\big], \tag{3}$$

where $\hat{y}$ are samples generated by randomly interpolating between $\tilde{y}$ and y, and $\mathbb{P}_{\hat{y}}$ is the distribution of $\hat{y}$.

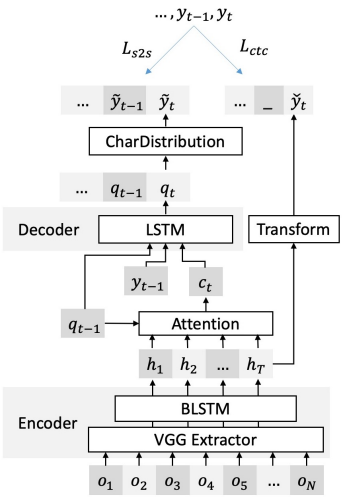


Fig. 3. Network architecture of the ASR model.

2.3. ASR Model

Network Architecture. Any network architecture for end-to-end speech recognition can be used here, while Fig. 3 gives the one used in this work, following the previous work [13] of integrating attentioned Seq2seq with CTC. The model takes a sequence of speech features $O = o_1, o_2, ..., o_N$ with length N as the input. O is encoded into sequence of hidden state $H = h_1, h_2, ..., h_T$ by the encoder (consists of a VGG extractor performing input downsampling followed by several BLSTM layers) with T being the output sequence length. The decoder is a single layer LSTM maintaining its own hidden state q. For each time index t, location-aware attention mechanism [7] $Attention$ is used to integrate H with the previous decoder state q_{t-1} to generate the context vector c_t. The decoder then decodes c_t together with the ground truth one-hot vector of the previous time step y_{t-1} to q_t. Finally, a fully connected layer with softmax activation $CharDistribution$ takes q_t and predicts the distribution vector $\tilde{y}_t$. h_t is also projected to $\check{y}_t$ with linear layer $Transform$ as output to help in learning of the encoder as shown in previous work [12]. The ASR model outputs two character sequences, $\tilde{y} = \tilde{y_1}, \tilde{y_2}, ..., \tilde{y_T}$ and $\check{y} = \check{y_1}, \check{y_2}, ..., \check{y_T}$, respectively supervised by Seq2seq loss and CTC loss. CLM only takes $\tilde{y}$ as input. During testing, $\tilde{y}$ and $\check{y}$ are integrated into a single output sequence just as done in the previous work [13].

Seq2seq Loss. The Seq2seq ASR model is to estimate the posterior probability,

$$P_{s2s}(\tilde{y}|O) = \prod_{l=1}^{T} P_{s2s}(\tilde{y}_l|\tilde{y}_{1:l-1}, O). \quad (4)$$

The loss function of the Seq2seq model model can be then computed as below,

$$L_{s2s} \equiv -\log P_{s2s}(y|O) = -\sum_{t=1}^{T} \log P_{s2s}(y_t|y_{1:t-1}, O), \quad (5)$$

except here $y = y_1, y_2, ..., y_T$ is the ground truth of O with length T.

CTC Loss. CTC [3] objective function is also used in this work for multi-task learning. CTC computes the posterior probability as below,

$$P(y|O) = \sum_{\pi \in y'} P(\pi|O), \quad (6)$$

where y' is the set of all possible sequences π obtained by arbitrarily repeating symbols of y and inserting blank symbols into y. The probability $P(\pi|O)$ can be approximated by $\check{y}$,

$$P(\pi|O) \approx \prod_{t=1}^{T} P_{ctc}(\check{y}_t|O). \quad (7)$$

The loss function of CTC is defined as:

$$L_{ctc} \equiv -\log P(y|O). \quad (8)$$

Total Loss. The ASR model is trained by minimizing the loss function constructed with Eq (5) and (8) minus the quality score from CLM,

$$L_{ASR} = \lambda_{s2s}\, L_{s2s} + (1 - \lambda_{s2s})\, L_{ctc} - \lambda_{CLM}\, CLM(\tilde{y}) \quad (9)$$

where λ_{s2s} controls the weights for the multi-task learning between Seq2seq and CTC. The last term in Eq (9) is the adversarial loss from CLM, pushing the ASR model to maximize the quality score from CLM.

3. EXPERIMENT

3.1. Experimental Setup

The experiments were performed on the LibriSpeech [27]. 100 hours of clean speech data and their transcriptions are used as the paired data. We took the text of other 360 hours of clean speech and 500 hours of noisy speech and utilized them as the unpaired data (text-only). The clean development set and clean test set were used for evaluation. We used the end-to-end speech processing toolkit ESPnet [28] for data preprocessing and customized it for our adversarial training processes. We followed the previous work [13, 21] to use 80-dimensional log Mel-filter bank and 3-dimensional pitch features as the acoustic features. Text data are represented by sequences of 5000 subword units one-hot vectors to avoid OOV. For the CLM model, the dimension of the output of all layers were set to 128 except the last. The first convolution had a window size of 2 and stride of 1, and the second had window size 3 and stride 1. Batch normalization is applied between layers. For the ASR model, the encoder included a 6-layer VGG extractor with downsampling used in the previous work [13] and a 5-layer BLSTM with 320 units per direction. 300-dimensional location-aware attention [7] was used in the attention layer. The decoder was a single layer LSTM with 320 units. λ_{gp} was set to 10 and λ_{s2s} is set to 0.5. λ_{CLM} is set to 10^{-4} since CLM output value was usually much higher than other loss values. Also, the update frequency of CLM is set to 5 times less than the ASR model to stabilize AT process.

Table 1. Speech recognition performance. "+LM" refers to shallow fusion decoding jointly with RNN-LM [13], "+AT" refers to the adversarial training proposed here, "+Both" indicates training with AT and joint decoding with RNN-LM, and BT is the prior work of back-translation [21].

Data	Method	CER/WER (%) Dev	CER/WER (%) Test	WER Δ† Test
(A) w/o unpair text	(a) Baseline	10.5 / 21.6	10.5 / 21.7	-
	(b) +LM	10.9 / 20.0	11.1 / 20.3	6.5%
	(c) +AT	**9.5 / 19.9**	**9.6 / 20.1**	**7.4%**
	(d) +Both	9.4 / 17.9	9.7 / 18.3	15.7%
(B) w/ 360hrs text	(e) +LM	10.5 / 19.6	10.6 / 19.6	9.7%
	(f) +AT	**9.1 / 19.1**	**9.5 / 19.2**	**11.5%**
	(g) +Both	9.0 / 17.1	9.1 / 17.3	20.3%
	(h) BT‡	10.3 / 23.5	10.3 / 23.6	6.3%
	(i) BT+LM‡	9.8 / 21.6	10.0 / 22.0	12.7%
	(j) Oracle§	3.6 / 8.1	3.6 / 8.2	62.2%
(C) w/ 860hrs text	(k) +LM	9.9 / 18.6	10.2 / 18.8	13.4%
	(l) +AT	**8.6 / 18.5**	**8.8 / 18.7**	**13.8%**
	(m) +Both	7.9 / 15.3	8.2 / 15.8	27.2%

† Relative improvement with respect to the baseline.
‡ Prior work [21], baseline WER 25.2% on test set reported.
§ Trained with 360hrs paired data and decoded with RNNLM.

3.2. Experimental Results

In the experiments, the ASR model was trained on the 100 hours speech data but combined with different amount of unpaired text utilized in different ways. The results are listed in Table 1, where "Baseline" refers to the plain end-to-end speech recognition framework as described in Sec. 2.3, "+LM" refers to the shallow fusion decoding with a separately trained RNN language model (RNN-LM) [13, 20] and "+AT" refers to the adversarial training proposed here. AT is actually compatible with any existing end-to-end speech recognition decoding approach, so "+Both" refers to training with AT while jointly decoding with RNN-LM. We ran all experiments three times with random initialization and reported the averaged error rate with decoding beam size set to 20.

Part (A) lists the results without extra text data. It is worth mentioning that even without extra text data, AT offered improvements over the baseline (rows(c) vs (a)), and the performance was further improved when integrated with RNN-LM (rows(d) vs (c)). Parts (B) and (C) are for results of different methods utilizing 360 hours and 860 hours of unpaired text data respectively. We see AT lowers recognition error rate as the RNN language model do (rows(f) vs (e), (l) vs (k)) and the improvements can be accumulated (rows(g) vs (f), (m) vs (l)). The previous work of back-translation (BT) style data augmentation [21], which aimed to utilize unpaired text data as AT do, was also listed in rows (h),(i). We see AT did better than BT under the same setting (rows (f) vs (h) and (g) vs (i)).

Fig 4 demonstrates the performance gap between different models (rows (a), (e), (f) and (g) of Table 1) with varying the beam size from 1 to 30. The points for beam size 20 are those in Table 1. We see that the proposed AT consistently

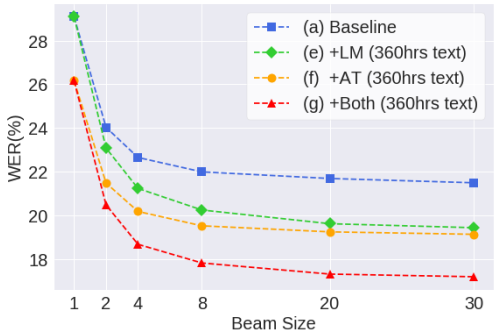


Fig. 4. Testing set WER for varying beam size. Index of curve shared with the corresponding row in Table 1.

improved the performance regardless of the beam size during decoding. It is clear that for all beam sizes considered AT outperformed RNN-LM in terms of utilizing the extra text data (curves (f) vs (e)), and AT is compatible to and able to offer additional improvements on top of the separately trained RNN-LM (curves (g) vs (f)). All these verified AT proposed here is able to integrate more linguistic knowledge from unpaired text data into the ASR model.

Table 2 provides some transcriptions obtained with the four models shown in rows (a)(e)(f)(g) of Table 1 on the same input utterances from the testing set. All models were trained with 100 hours of paired data, while the lower three with additional text of 360 hours, all with beam size 20. We see that AT seemed to make the output more grammatical. In the first example, AT is able to predict the correct words. In the second example, although all the models misrecognized the word "Alexander", the transcriptions provided by models with AT (rows (f)(g)) are more grammatical.

Table 2. Transcription examples, with ASR errors in uppercase and the differences made by AT underlined. Index shared with Table 1.

Model	Transcription
Truth	nonsense of course i can't really ...
(a) Baseline	NON SENSE of course i CAN'TVERLY ...
(e) +LM	NON SENSE of course i can't FREELY ...
(f) +AT	nonsense of course i can't really ...
(g) +Both	nonsense of course i can't really ...
Truth	alexander did not sit down
(a) Baseline	OUTSIDEED IT not SET down
(e) +LM	WHY did not sit down
(f) +AT	ALICE did not sit down
(g) +Both	ALICE did not sit down

4. CONCLUSION

In this paper we proposed a novel framework for adversarial training end-to-end speech recognition using a criticizing language model. This offers a direction for better utilizing additional text data without the need for a separately trained language model. This framework can be used with any end-to-end speech recognition and language modeling frameworks. Experiments on one example set of the proposed framework showed consistent improvement over different settings.

5. REFERENCES

[1] Frederick Jelinek, "Continuous speech recognition by statistical methods," *Proceedings of the IEEE*, vol. 64, no. 4, pp. 532–556, 1976.

[2] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, speech and signal processing (ICASSP)*. IEEE, 2013, pp. 6645–6649.

[3] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.

[4] Alex Graves and Navdeep Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International Conference on Machine Learning*, 2014, pp. 1764–1772.

[5] Hagen Soltau, Hank Liao, and Haim Sak, "Neural speech recognizer: Acoustic-to-word lstm model for large vocabulary speech recognition," in *Proc. Interspeech*, 2017, pp. 3707–3711.

[6] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4945–4949.

[7] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, "Attention-based models for speech recognition," in *Advances in neural information processing systems*, 2015, pp. 577–585.

[8] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.

[9] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, et al., "Deep speech 2: End-to-end speech recognition in english and mandarin," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, 2016, ICML'16, pp. 173–182.

[10] Rohit Prabhavalkar, Kanishka Rao, Tara N. Sainath, Bo Li, Leif Johnson, and Navdeep Jaitly, "A comparison of sequence-to-sequence models for speech recognition," in *Proc. Interspeech 2017*, 2017, pp. 939–943.

[11] Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, et al., "State-of-the-art speech recognition with sequence-to-sequence models," *ICASSP*, 2018.

[12] Suyoun Kim, Takaaki Hori, and Shinji Watanabe, "Joint ctc-attention based end-to-end speech recognition using multitask learning," in *Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 4835–4839.

[13] Takaaki Hori, Shinji Watanabe, Yu Zhang, and William Chan, "Advances in joint ctc-attention based end-to-end speech recognition with a deep cnn encoder and rnn-lm," in *Interspeech*, 2017.

[14] Samuel Thomas, Michael L Seltzer, Kenneth Church, and Hynek Hermansky, "Deep neural network features and semi-supervised training for low resource speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 6704–6708.

[15] Karel Vesely, Mirko Hannemann, and Lukas Burget, "Semi-supervised training of deep neural networks," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 267–272.

[16] Erinç Dikici and Murat Saraçlar, "Semi-supervised and unsupervised discriminative language model training for automatic speech recognition," *Speech Communication*, vol. 83, pp. 54–63, 2016.

[17] Shigeki Karita, Shinji Watanabe, Tomoharu Iwata, Atsunori Ogawa, and Marc Delcroix, "Semi-supervised end-to-end speech recognition," *Proc. Interspeech 2018*, pp. 2–6, 2018.

[18] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur, "Recurrent neural network based language model," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[19] Jan Chorowski and Navdeep Jaitly, "Towards better decoding and language model integration in sequence to sequence models," in *Proc. Interspeech 2017*, 2017, pp. 523–527.

[20] Anuroop Sriram, Heewoo Jun, Sanjeev Satheesh, and Adam Coates, "Cold fusion: Training seq2seq models together with language models," in *Interspeech*, 2018.

[21] Tomoki Hayashi, Shinji Watanabe, Yu Zhang, Tomoki Toda, Takaaki Hori, Ramon Astudillo, and Kazuya Takeda, "Back-translation-style data augmentation for end-to-end asr," *arXiv preprint arXiv:1807.10893*, 2018.

[22] Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura, "Listening while speaking: Speech chain by deep learning," in *Automatic Speech Recognition and Understanding Workshop (ASRU), 2017 IEEE*. IEEE, 2017, pp. 301–308.

[23] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[24] Martin Arjovsky, Soumith Chintala, and Léon Bottou, "Wasserstein generative adversarial networks," in *International Conference on Machine Learning*, 2017, pp. 214–223.

[25] Cédric Villani, *Optimal transport: old and new*, vol. 338, Springer Science & Business Media, 2008.

[26] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville, "Improved training of wasserstein gans," in *Advances in Neural Information Processing Systems*, 2017, pp. 5767–5777.

[27] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[28] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai, "Espnet: End-to-end speech processing toolkit," in *Interspeech*, 2018.