INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

General-purpose Adversarial Training for Enhanced Automatic Speech Recognition Model Generalization

Dohee Kim[*], *Daeyeol Shim*[*†] *and Joon-Hyuk Chang*[*]

[*]Hanynag University, Seoul, Republic of Korea
[†]Amazon Web Services
{dohe0342, shimdx, jchang}@hanyang.ac.kr

Abstract

We present a new adversarial training method called General-purpose adversarial training (GPAT) that enhances the performance of automatic speech recognition models. In GPAT, we propose the followings: (1) a plausible adversarial examples converter (PAC); (2) a distribution matching regularization term (DM reg.). Compared to previous studies that directly compute gradients with respect to the input, PAC incorporates non-linearity to achieve performance improvement while eliminating the need for extra forward passes. Furthermore, unlike previous studies that use fixed norms, GPAT can generate similar yet diverse samples through DM reg. We demonstrate that the GPAT elevates the performance of various models on the LibriSpeech dataset. Specifically, by applying GPAT to the conformer model, we achieved 5.3% average relative improvements. With respect to the wav2vec 2.0 experiments, our method yielded a 2.0%/4.4% word error rate on the LibriSpeech test sets without a language model.

Index Terms: speech recognition, adversarial training, data augmentation

1. Introduction

Automatic speech recognition (ASR) has made significant progress thanks to the development of model structures such as transformer [1] and conformer [2], and pre-training strategies such as self-supervised learning (SSL) [3–6]. However, it is important to note that deep-learning based models tend to overfit quickly and require a considerable amount of training data [7]. To overcome this issue and ensure high accuracy, several data augmentation techniques have been proposed. For instance, time-domain augmentations [8] modify the sampling rate of an input, whereas simulation-based data augmentation involves adding noise [9], using a room impulse response function to simulate point sources spread in space [10], or vocal tract length perturbation [11]. Moreover, SpecAugment [12], which demonstrates promising performance, has been proposed by masking the time and frequency axes of a mel spectrogram. Additionally, synthetic data and adversarial examples have been suggested for data augmentation [13–21].

Meanwhile, several studies have been conducted to enhance the generalization of models using adversarial examples in various domains. [14, 15] improved image classification and object detection using adversarial examples. [16] applied adversarial training to natural language processing (NLP) and achieved better accuracy. [19] proposed virtual adversarial training (VAT) [20] can improve ASR performance. [21] also demonstrated that adversarial training can improve ASR performance in accented speech. The previous studies demonstrated that adversarial training can enhance performance in various domains. However, in order to perform a single update through adversarial training, an extra forward pass is necessary to obtain adversarial examples, resulting in a increase in training time. Additionally, obtaining adversarial examples requires a direct calculation of gradients with respect to the inputs, which limits the potential for performance enhancement [22] and fixed norms are utilized in order to obtain adversarial examples which is not suitable for the sequential data.

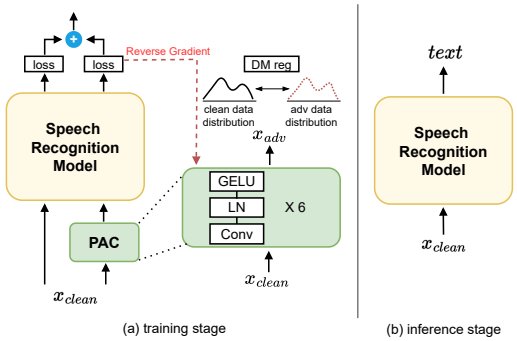

Figure 1: *Overview of the proposed method. (a) In the training stage, PAC transforms clean examples into adversarial examples with similar distribution thanks to the DM reg. The target network learns clean and adversarial examples both. We train the target network and the PAC simultaneously. (b) At the test step, PAC is dropped and we only use the recognition model.*

To address this issue, we propose general-purpose adversarial training (GPAT), a simple method that can utilize adversarial examples for data augmentation to enhance the generalization of the ASR models. As shown in Figure 1, GPAT has two novel components: (1) a plausible adversarial example converter (PAC) that converts clean examples into adversarial ones and feeds them as training data for the ASR model; (2) a distribution-matching regularization term (DM reg.) that is designed to alleviate the distribution mismatch problem [14]. Compared to previous studies that obtained adversarial examples by directly calculating gradients with respect to the inputs, GPAT utilizes PAC with non-linearity, enabling further enhanced performance. Moreover, generating adversarial examples through the use of PAC does not require repetitive forward passes. Additionally, by utilizing DM reg., distribution mismatch problem between adversarial and clean examples [14] addressed, thus allowing for stable training.

We apply GPAT to self-supervised models and attention-based encoder-decoder (AED), demonstrating improved performance through adversarial training with GPAT in general situations. For example, AED model trained with GPAT achieved

[†]This work is not related to Amazon Web Services

10.21437/Interspeech.2023-2389

2.2%/5.0% word error rate (WER), beating its vanilla counterpart by 4.5%/3.8% relative reduction on the LibriSpeech [23] test-clean and test-other subsets. The improvement achieved by GPAT was more notable when applied to the self-supervised models. GPAT helped wav2vec 2.0 BASE gain a relative average improvement of 7.7 % on the LibriSpeech two subsets.

2. Adversarial training

Adversarial training, which trains the target network with adversarial examples, usually focuses on defending against adversarial attacks [24]. However, recent studies have shown that adversarial examples are useful for data augmentation and enhance model performance. For example, [14] showed that adversarial examples with a similar distribution to clean ones can improve model generalization. [16] applied adversarial training to NLP and achieved better standard accuracy. Moreover, adversarial training for data augmentation in ASR is usually studied for specific situations. [17] showed that adversarial training can bring better accuracy to disordered speech. Other studies have demonstrated that adversarial training is helpful in situations with noisy [18] or accented [21] speech. In general situations, the application of VAT as a form of data augmentation by [19] led to improved ASR performance.

However, previous studies have required the computation of gradients with respect to the input in order to obtain adversarial examples. Consequently, obtaining adversarial examples requires additional forward passes, leading to a significant increase in training time. Additionally, as direct gradient calculation with respect to the input lacks non-linearity, the potential for performance improvement may be limited [22]. Additionally, adversarial noise that is added to clean input is unsuitable for the sequential data due to the utilization of a fixed norm.

3. Proposed method

We introduce GPAT as a method of achieving improved performance without the need for additional forward passes in adversarial training. This framework comprises two key components: (1) PAC, which transforms clean examples into adversarial examples, and (2) DM reg., a regularization term that facilitates the generation of adversarial examples by the PAC while preserving a similar distribution as clean examples. In Sec. 3.1, we present the conventional adversarial training method, VAT, while Sec. 3.2 delves into the PAC. In Sec. 3.3, we introduce the training of the PAC with DM reg.

3.1. Virtual adversarial training

Virtual Adversarial Training (VAT) is a technique that calculates adversarial perturbations (also known as noise) represented by the vector r_{adv}, which is applied to the input x_c to create an adversarially enriched input x_{adv}. The selection of r_{adv} involves a small perturbation in the direction that maximally increases the loss of the model. The magnitude of this adversarial noise is a hyperparameter. The VAT regularization loss is determined by the Kullback–Leibler divergence (KLD) [25] between the predictions.

$$\mathcal{R}_{VAT} = KLD(p(y|x_c)||p(y|x_c + r_{adv})). \quad (1)$$

When applying VAT to the sequential data in [19], the noise tensor r_{adv} has the same dimensions as x_c. To ensure the stability of the gradient used in the generation of r_{adv}, they calculated it with respect to the teacher-forced training of the target variable y. However, direct calculation of gradients with respect to the inputs limits the potential for performance enhancement. And since they compute the adversarial noise while keeping the norm fixed, the intensity of the noise would vary depending on the sequential data. Furthermore, an additional forward pass is required to compute the adversarial noise.

Table 1: *WER on the Librispeech test-clean/test-other of wav2vec 2.0 BASE with or without GPAT. Both models trained on LibriSpeech 960 h labeled dataset.*

Settings	wav2vec 2.0 BASE	wav2vec 2.0 BASE + GPAT
WER(%)	3.40 / 8.56	3.21 / 8.23

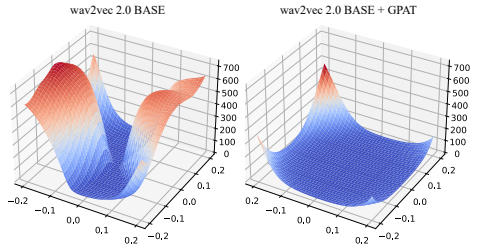

Figure 2: *Loss landscape of wav2vec 2.0 BASE with or without GPAT on LirbriSpeech 960 h. The models are trained on LibriSpeech 960 h labeled dataset. Visualization tools are provided by [26]*

3.2. Plausible adversarial examples converter

To eliminate redundant forward passes and introducing non-linearity, we propose a plausible adversarial examples converter (PAC). PAC is a simple network that transforms clean examples into adversarial ones. As illustrated in Figure 1, the PAC is composed of six blocks, where the block module consists of 1d-Conv, layer-norm, and GELU activation. There are a negligible number of parameters for network training, for example, 0.1% more parameters than the baseline on wav2vec 2.0 LARGE. In the test step, this additional auxiliary network is dropped and we only use the baseline model for inference. Compared to the conventional method of directly computing gradients with respect to the input, PAC incorporates non-linearity to generate adversarial examples, resulting in greater performance improvement. Additionally, the PAC is capable of generating adversarial examples without requiring any additional forward passes on the target network.

3.3. Distribution-matching regularization

We improve the generalization of ASR models by extending the adversarial loss with a new distribution-matching regularization term denoted by DM reg. As previously mentioned, the conventional approach requires a hyperparameter to determine the norm of the adversarial noise, which remains fixed regardless of the length of the sequential data (e.g., time), resulting in varying noise intensity. To address this issue, we use DM reg. to ensure that the adversarial examples generated by PAC are implicitly similar to clean examples. The term is formulated as follows:

$$\mathcal{R}_{DM} = \frac{1}{NT} \sum_{n=1}^{N} \sum_{t=1}^{T} |PAC(x_c^{nt}) - x_c^{nt}|_2^2 \quad (2)$$

where x_c^{nt} represents the features for each sample n and each time step t. In addition, N denotes the sample number of the mini-batch and T is the total time step of each sample. Since

the role of DM reg. is to make the adversarial examples similar to clean ones, it can also be achieved using KLD.

The PAC should generate hard samples for the target network with a similar distribution to that of clean samples. Hence, the final loss function of the PAC, including DM reg., can be expressed as follows:

$$\mathcal{L}_{PAC} = -\mathcal{L}_{target} + \alpha \mathcal{R}_{DM} \tag{3}$$

where $\mathcal{L}_{target}$ is the target network loss function such as connectionist temporal classification (CTC) loss [27]. Also, α is a hyperparameter to scale the adversarial loss and DM reg.

As shown in Table 1 and Figure 2, the model trained with GPAT obtained a flatter loss landscape and low WER on LibriSpeech dataset, which is commonly used for estimating the general performance. These results exhibited that GPAT allows the target model to smooth out diverse directions of the decision boundary. GPAT algorithm is summarized in Algorithm 1.

Algorithm 1: Pseudo code of adversarial training using GPAT

Data: A set of clean audio with labels;
Result: Network params θ;
for *each training step* **do**
  Sample a clean audio mini-batch x_c with label y;
  Convert clean batch into adversarial batch x_a
  $x_a \leftarrow PAC(x_c; \theta_{pac})$
  Compute DM reg. according to Eq. 2.
  Compute $\mathcal{L}_c(\theta, x_c, y)$ on clean mini-batch
  Compute $\mathcal{L}_a(\theta, \theta_{pac}, x_a, y)$ on adversarial batch
  Minimize the target loss w.r.t network params
  $\arg\min_{\theta} \mathcal{L}_c(\theta, x_a, y) + \mathcal{L}_a(\theta, x_c, y)$
  Minimize the PAC loss in Eq. 3. w.r.t PAC params
  $\arg\min_{\theta_{pac}} -\mathcal{L}_a(\theta_{pac}, x_a, y) + \alpha \mathcal{R}_{\mathcal{DM}}(\theta_{pac}, x_a, x_c)$.
end
return θ

4. Experiments

To verify GPAT can enhance the ASR model generalization, we used the LibriSpeech dataset, which contains 960 h of speech from public-domain audiobooks. We applied GPAT to attention-based encoder decoder (AED) and self-supervised models. Also, we trained for 100 h and 960 h to show that GPAT can be applied regardless of the amount of data. We utilized the implementation from ESPNet [28] for the AED models. We employed the self-supervised model implementations from fairseq [29] and UniLM [30]. For a fair comparison, we adopted the same hyperparameters with or without GPAT throughout the training procedure and followed each recipes setting. All the ASR models were evaluated without a language model. We presented GPAT results in Sec. 4.1. We presented the analysis, including comparisons with other augmentation methods, data visualization, and ablation study, in Sec. 4.2. Experiments with 100 h dataset were performed on 4 NVIDIA A100 40GB GPUs and large dataset were performed on Amazon SageMaker using ml.p4d.24xlarge instance.

4.1. Results of the LibriSpeech corpus

4.1.1. Attention-based encoder decoder model results

First, we studied the effectiveness of GPAT in the AED models. We adopted 18-layer transformer-based and conformer-based encoders along with a 6-layer transformer-based decoder. We utilized 80 dimension log mel spectrogram features with delta and double-delta stacking. Setting the encoder dimension to 512, we applied SpecAugment [12] and speed perturbation [31] to all speech samples. The PAC was placed next to the log mel spectrogram. Therefore, the PAC receives the log mel spectrograms as input and provides difficult input features to the target network. We tokenized label sentences as 500 subwords with sentencepeice [32]. The model checkpoint for each epoch was saved, and the final model was produced by averaging 10 checkpoints with the best validation accuracy. For training stability, a near-identity initialization is required for the PAC. Thus, we first ran 10 epochs for 100 h and 1 epoch for 960 h to the PAC using the DM reg only. We chose 10^3 as the scale parameter α. Note that α matches the scale between the two loss terms. We used a large number as a scale parameter because of the scale gap between the adversarial loss and DM reg. We trained the PAC using Adam [33] with a learning rate of 0.001. The WERs of the recognition models are reported in Table 2 on the two official dev and test sets. We achieved superior performance compared with the baseline. Especially in 100 h training, we obtained an absolute WER reduction of 1.9% on the transformer. For LibriSpeech 960 h in conformer based model, we obtained an average performance improvement of 5.3%.

Table 2: *WER on the LibriSpeech dev/test sets without language model when applying GPAT method.*

Method	train dataset	dev		test		Relative
		clean	other	clean	other	
Baseline						
transformer	100 h	10.8	25.0	11.3	25.3	-
	960 h	4.8	11.2	5.0	10.9	-
conformer	100 h	8.3	21.4	8.6	22.0	-
	960 h	2.2	5.4	2.3	5.2	-
This work						
transformer	100 h	9.7	23.3	9.9	23.4	9.8 %
	960 h	4.6	10.9	4.7	10.6	3.4 %
conformer	100 h	7.6	20.4	8.1	20.8	5.6 %
	960 h	2.0	5.1	2.2	5.0	5.3 %

Table 3: *WER on the LibriSpeech dev/test sets without language model when applying GPAT method. * denotes our re-implementations*

Method	fine-tuning dataset	dev		test		Relative
		clean	other	clean	other	
Baseline						
wav2vec 2.0 Base	100 h	6.1	13.5	6.1	13.3	-
	960 h	3.2	8.9	3.4	8.5	-
wav2vec 2.0 Large	100 h	3.3	6.5	3.1	6.3	-
	960 h	2.1	4.5	2.2	4.5	-
HuBERT Base*	100 h	5.4	12.8	5.4	12.5	-
WavLM Base*	100 h	5.9	14.5	6.1	14.3	-
data2vec Base	100 h	4.2	9.6	4.2	9.7	-
This work						
wav2vec 2.0 Base	100 h	5.4	12.9	5.3	12.4	7.7 %
	960 h	3.0	8.2	3.1	8.2	6.3 %
wav2vec 2.0 Large	100 h	3.1	6.3	3.1	6.1	3.1 %
	960 h	2.0	4.3	2.0	4.4	4.5 %
HuBERT Base	100 h	5.1	12.3	5.2	12.1	3.9 %
WavLM Base	100 h	5.4	13.9	5.6	13.9	4.9 %
data2vec Base	100 h	4.0	9.3	4.1	9.3	3.6 %

4.1.2. Self-supervised Model Results

We adopted GPAT for self-supervised models, which have been studied extensively recently. We followed the hyperparameter settings of each paper and performed fine-tuning with CTC loss on a pre-trained model. We tokenized label sentences as characters. We utilized the best validation accuracy model as the final model. The architecture of the self-supervised model typically consists of a feature extractor (e.g., a CNN) and an encoder (e.g., a transformer). As the feature extractor was not trained during fine-tuning, its output was the actual input in the fine-tuning stage. Therefore, we placed the PAC next to the feature extractor, and the remaining settings were the same as those in

Table 4: *Comparison with other augmentation methods. We utilized LibriSpeech 100 h fine-tuned wav2vec2 BASE model.*

Method	dev		test		Relative	Train time (h)
	clean	other	clean	other		
Baseline(No aug)	6.0	15.0	6.4	14.9	-	11.8
+SpecAug	6.1	13.5	6.1	13.3	7.8 %	11.8
+RIR	6.0	14.8	6.3	14.7	1.1 %	23.1
+Speed	5.8	14.6	6.3	14.6	2.4 %	34.3
+SpecAug+VAT	5.8	13.5	5.8	13.0	9.9 %	28.6
+SpecAug+GPAT	**5.4**	**12.9**	**5.3**	**12.4**	**14.9** %	25.1

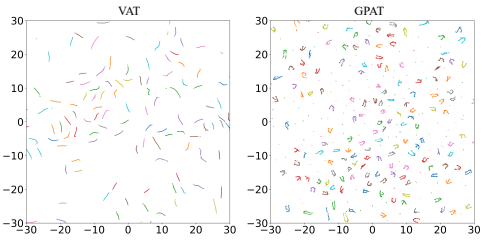

Figure 3: *Visualization of augmented samples. Since VAT uses fixed norm noise to generate adversarial examples, it generates monotonous samples. On the other hand, GPAT, thanks to the PAC and DM reg., generates diverse examples, but similar.*

the end-to-end model training. We adopted GPAT for various self-supervised models, such as wav2vec 2.0 [3], HuBERT [4], WavLM [5], and data2vec [6]. As shown in Table 3, trained with GPAT yielded noteworthy results for all the self-supervised models. Note that we used the same hyperparameters described in the respective papers, which indicates that GPAT with appropriate hyperparameters can improve performance further. In particular, GPAT increased the data2vec performance, which yielded the best performance for the same parameters. This demonstrates that the proposed scheme can further enhance the state-of-the-art performance.

4.2. Analysis

4.2.1. Comparison with other augmentation methods

To demonstrate that our method achieves superior performance improvement and enables faster training compared to other augmentation methods, we conducted comparative experiments with other data augmentations and adversarial training techniques. We conducted experiments on wav2vec2 BASE 100 h fine-tuning with various augmentation methods. We trained 221 epochs and other hyperparameters for the experiments were set as shown in [3]. As depicted in Table 4, conventional data augmentation methods such as SpecAug, RIR, and speed perturbation, except for SpecAug, were found to be ineffective in improving generalization performance. The training time increased proportionally with the amount of data. VAT with SpecAugment showed some performance improvement, it did not achieve significant improvement because it used a fixed norm for r_{adv} without considering the length of the sequential data and directly computed gradients with respect to the input without incorporating non-linearity. In contrast, our method showed performance improvement by using DM reg. to implicitly make adversarial examples similar to clean ones and incorporating non-linearity with the PAC. Although the addition of the PAC required extra time, we achieved shorter training time compared to VAT by eliminating redundant forward passes.

4.2.2. Visualization

To demonstrate that our method can generate more diverse adversarial examples compared fixed norm noise used in VAT, we visualized the adversarial augmented samples using t-SNE [34]. We used a wav2vec2 BASE 100 h fine-tuned model for the experiments, and since the adversarial examples were generated in every epoch, we visualized all generated data. We randomly sampled 200 utterances from the training set and performed t-sne after padding to match the shape of the data to the longest utterance. Since VAT uses fixed norm noise, it generated data on a fixed distance sphere. As a result, Figure 3 exhibited that VAT generated monotonous samples. In contrast, our method generated diverse samples by not limiting adversarial noise with a fixed norm and using DM reg. to make adversarial examples similar to clean ones. This makes the decision boundary more distinguishable, resulting in a flatter loss landscape as shown in Figure 2 and better performance improvement.

Table 5: *Effectiveness of each component of our method. The model trained with all components of GPAT has the best performance, especially on test sets.*

Adv loss	DM reg.	dev		test		Relative
		clean	other	clean	other	
Baseline		6.1	13.5	6.1	13.3	-
✓		6.2	14.0	6.4	13.7	-3.3 %
	✓	5.9	13.2	5.8	12.9	3.1 %
✓	✓	**5.4**	**12.9**	**5.3**	**12.4**	7.8 %

4.2.3. Ablation Study

To isolate the effects of the PAC and DM reg., we conducted an ablation study as shown in Table 5. We fine-tuned the wav2vec 2.0 BASE model using the LibriSpeech 100 h subset. As shown in Table 5, using only adversarial loss, we could not achieve performance improvement because distribution matching was not guaranteed. When using only DM reg., we could improve the model performance. However, we observed that without adversarial loss, the output of the PAC was the same as the input after a few epochs. Thus, the performance improvement was lower than that of GPAT because of the lack of diversity. We could boost the recognition model by employing adversarial loss and DM reg. In summary, GPAT with the addition of non-linearity outperformed other methods and achieved the highest level of performance without using fixed norm noise.

5. Conclusion

Previous studies have commonly used adversarial examples for specific situations. However, they generated adversarial examples using fixed norm, which is not suitable for the sequential data, along with additional forward passes. Moreover, the absence of non-linearity in generating adversarial examples constrained the enhancements in performance. Here, we offer a different perspective: by using the PAC, we introduced non-linearity and eliminated the need for additional forward passes, enabling faster learning, and by using DM reg., we generated adversarial examples that are similar to clean samples without fixed norm. Through extensive experiments, we demonstrated that GPAT can be applied not only to AED models but also to recently proposed self-supervised models. Moreover, we confirmed that our method achieves significant performance improvement compared to other augmentation methods and generates similar yet diverse samples through visualization.

Acknowledgement This work was supported by Institute of Information communications Technology Planning Evaluation (IITP) grant funded by the Korea government(MSIT) (No. 2017-0-00474, Intelligent Signal Processing for AI Speaker Voice Guardian)

6. References

[1] Y. Wang, A. Mohamed, D. Le, C. Liu, A. Xiao, J. Mahadeokar, H. Huang, A. Tjandra, X. Zhang, F. Zhang *et al.*, "Transformer-based acoustic modeling for hybrid speech recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing* (ICASSP), 2020, pp. 6874–6878.

[2] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented Transformer for Speech Recognition," in *Proc. Interspeech*, 2020, pp. 5036–5040.

[3] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. Advances in Neural Information Processing Systems* (NeurIPS), vol. 33, 2020, pp. 12 449–12 460.

[4] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units," in *Proc. IEEE/ACM Transactions on Audio, Speech, and Language Processing* (TASLP), vol. 29, 2021, pp. 3451–3460.

[5] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, "WavLM: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, 2022.

[6] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli, "Data2vec: A general framework for self-supervised learning in speech, vision and language," *arXiv preprint arXiv:2202.03555*, 2022.

[7] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing* (ICASSP), 2018, pp. 4774–4778.

[8] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Sixteenth annual conference of the international speech communication association*, 2015.

[9] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates *et al.*, "Deep speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[10] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing* (ICASSP), 2017, pp. 5220–5224.

[11] N. Jaitly and G. E. Hinton, "Vocal tract length perturbation (VTLP) improves speech recognition," in *Proc. ICML Workshop on Deep Learning for Audio, Speech and Language*, vol. 117, 2013, p. 21.

[12] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[13] T.-Y. Hu, M. Armandpour, A. Shrivastava, J.-H. R. Chang, H. Koppula, and O. Tuzel, "SYNT++: Utilizing Imperfect Synthetic Data to Improve Speech Recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing* (ICASSP).

[14] C. Xie, M. Tan, B. Gong, J. Wang, A. L. Yuille, and Q. V. Le, "Adversarial examples improve image recognition," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition* (CVPR), 2020, pp. 819–828.

[15] T. Chen, Y. Cheng, Z. Gan, J. Wang, L. Wang, Z. Wang, and J. Liu, "Adversarial feature augmentation and normalization for visual recognition," *arXiv preprint arXiv:2103.12171*, 2021.

[16] D. Wang, C. Gong, and Q. Liu, "Improving neural language modeling via adversarial training," in *Proc. International Conference on Machine Learning* (ICML), 2019, pp. 6555–6565.

[17] Z. Jin, M. Geng, X. Xie, J. Yu, S. Liu, X. Liu, and H. Meng, "Adversarial data augmentation for disordered speech recognition," *arXiv preprint arXiv:2108.00899*, 2021.

[18] B. Liu, S. Nie, S. Liang, W. Liu, M. Yu, L. Chen, S. Peng, C. Li *et al.*, "Jointly Adversarial Enhancement Training for Robust End-to-End Speech Recognition." in *Proc. Interspeech*, 2019, pp. 491–495.

[19] G. Wang, A. Rosenberg, Z. Chen, Y. Zhang, B. Ramabhadran, and P. J. Moreno, "Scada: Stochastic, consistent and adversarial data augmentation to improve asr." in *INTERSPEECH*, 2020, pp. 2832–2836.

[20] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: a regularization method for supervised and semi-supervised learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1979–1993, 2018.

[21] N. Das, S. Bodapati, M. Sunkara, S. Srinivasan, and D. H. Chau, "Best of both worlds: Robust accented speech recognition with adversarial transfer learning," *arXiv preprint arXiv:2103.05834*, 2021.

[22] Y. Kim, D. Park, D. Kim, and S. Kim, "Naturalinversion: Data-free image synthesis improving real-world consistency," in *Proc. AAAI*, vol. 36, no. 1, 2022, pp. 1201–1209.

[23] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing* (ICASSP).

[24] H. Xu, Y. Ma, H.-C. Liu, D. Deb, H. Liu, J.-L. Tang, and A. K. Jain, "Adversarial attacks and defenses in images, graphs and text: A review," *International Journal of Automation and Computing*, vol. 17, no. 2, pp. 151–178, 2020.

[25] T. M. Cover, *Elements of information theory*. John Wiley & Sons, 1999.

[26] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the loss landscape of neural nets," in *Proc. Advances in neural information processing systems* (NeurIPS), vol. 31, 2018.

[27] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proc. International Conference on Machine Learning* (ICML), 2006, pp. 369–376.

[28] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen *et al.*, "ESPnet: End-to-end speech processing toolkit," *arXiv preprint arXiv:1804.00015*, 2018.

[29] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, "fairseq: A fast, extensible toolkit for sequence modeling," *arXiv preprint arXiv:1904.01038*, 2019.

[30] "Unilm," https://github.com/microsoft/unilm.

[31] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Sixteenth annual conference of the international speech communication association*, 2015.

[32] T. Kudo and J. Richardson, "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," *arXiv preprint arXiv:1808.06226*, 2018.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[34] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE." *Journal of machine learning research*, vol. 9, no. 11, 2008.