Interspeech 2022
18-22 September 2022, Incheon, Korea

Chunking Defense for Adversarial Attacks on ASR

Yiwen Shao, Jesús Villalba, Sonal Joshi, Saurabh Kataria, Najim Dehak, Sanjeev Khudanpur

Center of Language and Speech Processing,
The Johns Hopkins University, Baltimore, MD 20218, USA
{yshao18,jvillal7,sjoshi12,skatari1,ndehak3,khudanpur}@jhu.edu

Abstract

While deep learning has lead to dramatic improvements in automatic speech recognition (ASR) systems in the past few years, it has also made them vulnerable to adversarial attacks. These attacks may be designed to either make ASR fail in producing the correct transcription or worse, output an adversary-chosen sentence. In this work, we propose a defense based on independently processing random or fixed size chunks of the speech input in the hope of "containing" the cumulative effect of the adversarial perturbations. This approach does not require any additional training of the ASR system, or any defensive pre-processing of the input. It can be easily applied to any ASR systems with little loss in performance under benign conditions, while improving adversarial robustness. We perform experiments on the Librispeech data set with different adversarial attack budgets, and show that the proposed defense achieves consistent improvement on two different ASR systems/models.
Index Terms: speech recognition, adversarial attack and defense, adversarial robustness, streaming model

1. Introduction

With the rapid development of deep learning techniques in recent years, more and more neural-based AI systems have been deployed in real world scenarios. Among them, the automatic speech recognition (ASR) is one of the most successful and widely used application [1, 2, 3]. However, there is a rising concern that neural ASR systems can be easily manipulated by adversarial inputs with imperceptible distortions [4, 5].

Depending on how much knowledge the adversary has about the deep learning system, adversarial attacks are classified as either black-box or white-box attacks. In the black-box scenarios, the attacker has no access to the ASR system/model but is allowed to "probe" it with manipulated inputs and observing its input. These attacks are thus harder to mount and, consequently, less effective than white-box attacks. As its name suggests, white-box attacks can get access to all the information about the ASR system, including its model architecture, parameters and training data. Among all proposed attacks, the fast gradient sign method (FGSM) [4] and its multi-step iterative version, called the projected gradient descent (PGD) attack [5], remain the most successful attacks in many fields of deep learning, including ASR. Although there are also some newly developed attacks specific to ASR [6, 7], their effectiveness is reported to be less consistent and robust on strong ASR models [8]. We focus on the white-box threat model in this paper, and study defenses against PGD attacks with different attack budgets and iterative steps.

Several efforts have been made to defend ASR systems against adversarial attacks. In [8, 9], pre-processing modules are added to the ASR system that attempt to recover the benign audio signal from the adversarial input. However, these pre-processing modules are vulnerable to adaptive attacks too, i.e., white-box attacks that propagate the gradient through both the ASR model and the pre-processing modules. By contrast, adversarial training [5, 10] can achieve consistent robustness against these attacks by feeding the ASR model with adversarial examples during training. But this has two limitations: (1) Generating adversarial examples on-the-fly during training multiplies training time $\times(N+1)$, where N is the number of iterations (of back-propagation through the ASR neural network) needed to create the adversarial version of each training example; (2) Adversarial training shifts the data distribution from benign examples to adversarial examples, resulting in non-negligible loss of performance in benign conditions.

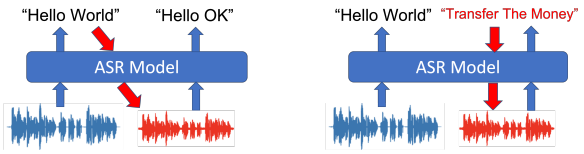


Figure 1: *Illustration of an* untargeted *(left) and a* targeted *attack (right), in which an adversary manipulates a benign (blue) signal to generate adversarial (red) signals that fail to produce the correct output (left), or produce a desired incorrect output (right).*

In summary, most of existing approaches focus on solving the distribution mismatch between benign and adversarial data in such problems, but very few attempt to utilize the sequential nature of ASR to create defenses. Temporal dependencies within speech are used in [11] to characterize adversarial examples, but only to detect attacks.

Contributions: We develop a new defense, named chunking defense, against adversarial attacks on sequence-to-sequence tasks like ASR. Instead of defending against adversarial attacks, we "counterattack" by cutting up a sequence into multiple shorter sequences or *chunks*. The proposed defense can be applied to any existing ASR model, without extra pre-processing modules or adversarial training/fine-tuning of the model. Experimental results on two state-of-the-art ASR models, (i) an E2E LF-MMI model [12, 13], and (ii) a streaming CTC model, both built using K2 [14], show that the proposed defense achieves significant robustness compared to the undefended models, especially against targeted attacks.

2. Adversarial Attacks on ASR

Given an utterance $\mathbf{X_t}$, the output prediction $\mathbf{P_t}$ of a neural net based ASR model is obtained by:

$$\mathbf{P_t} = F(\mathbf{X_t}; \theta) \tag{1}$$

where $\mathbf{t}$ is a time interval (t_s, t_e). $F(\cdot;\theta)$ represents the forward function of the neural network with parameter θ.

10.21437/Interspeech.2022-11096

In an adversarial setting, an attacker finds the perturbation ${\sigma_{\mathbf{t}}}^{\text{adv}}$ within a permissible set S to create an adversarial utterance ${\mathbf{X_t}}^{adv} = \mathbf{X_t} + {\sigma_{\mathbf{t}}}^{adv}$ that either (a) maximizes the loss between the ground truth $\mathbf{Y_t}$ and the prediction ${\mathbf{P_t}}^{adv} = F(\mathbf{X_t^{adv}}; \theta)$, i.e. that aims to make the system's output transcription go wrong, as illustrated in Fig. 1 left, so that

$$ {\sigma_{\mathbf{t}}}^{adv} = \arg\max_{\sigma_{\mathbf{t}} \in S} L(\mathbf{P_t^{adv}}, \mathbf{Y_t}), \tag{2} $$

or (b) minimizes the loss between a target sentence ${\mathbf{Y_t}}^{tar}$ and the prediction ${\mathbf{P_t}}^{adv}$, i.e. that tries to make the output be the target sentence as illustrated in Fig. 1 right, so that

$$ {\sigma_{\mathbf{t}}}^{adv} = \arg\min_{\sigma_{\mathbf{t}} \in S} L({\mathbf{P_t}}^{adv}, {\mathbf{Y_t}}^{tar}). \tag{3} $$

Typically $S = \{{\sigma_{\mathbf{t}}}^{adv} : \|{\sigma_{\mathbf{t}}}^{adv}\|_p \leq \varepsilon\}$ for some p and ε.

Fast gradient sign method (FGSM) [4] takes the sign of the gradient of the loss L in (2) or (3) w.r.t. the input $\mathbf{X_t}$ to get

$$ {\sigma_{\mathbf{t}}}^{adv} = \pm\varepsilon \,\text{sign}(\nabla_{\mathbf{X_t}} L(\cdot)) \tag{4} $$

where ε is the attack budget that restricts $\|{\sigma_{\mathbf{t}}}^{adv}\|_\infty \leq \varepsilon$. Note that $+$ is for untargeted attack that represents the direction mamximizing $L(\cdot)$ in (2)) while $-$ is for targeted attack that minimizing $L(\cdot)$ in (3).

Projected Gradient Descent (PGD) [5] is an iterative version of FGSM that takes small steps α in the direction of the gradient while clipping the perturbation to stay within budget:

$$ \begin{aligned} {\mathbf{X_t}}^{(i+1)} &= \texttt{clip}_{\mathbf{X_t} \pm \varepsilon}\left({\mathbf{X_t}}^{(i)} \pm \alpha \,\text{sign}(\nabla_{{\mathbf{X_t}}^{(i)}} L(\cdot)\right), \\ {\mathbf{X_t}}^{adv} &= {\mathbf{X_t}}^{(I)}, \end{aligned} \tag{5} $$

where $1 \leq i \leq I$, α is the step size, and I is the number of iterations, both important factors determining the strength of the attack. For a given budget ε, more iterations with smaller steps lead to stronger attacks, while for $I = 1$, PGD is equivalent to FGSM with $\varepsilon = \alpha$.

Untargeted attacks are not hard to conduct on an undefended ASR system, as easily seen in both our experiments and previous work [15, 8]. FGSM, PGD with few iterations or even Gaussian noise can cause such degradation on an ASR model trained with clean speech. On the other hand, it takes more effort to mount a successful targeted attack where the WER w.r.t the target sentence ${\mathbf{Y_t}}^{tar}$, denoted WER_{TGT}, is very low, e.g. comparable to WER_{REF} on benign inputs. In real life, malicious targeted attacks have the potential to cause more harm than untargeted attacks. Defending against targeted attack is therefore our primary concern in this work.

Several targeted attacks on ASR systems have been proposed in the literature, including the C&W attack [6], imperceptible attack [7], and PGD attacks with large number of iterations. We follow the current best practice developed by the DARPA GARD[1] Evaluation team, who report that performance against high-iteration PGD attacks is indicative of wider adversarial robustness.

3. Chunking to Defuse Targeted Attacks

3.1. Partition function

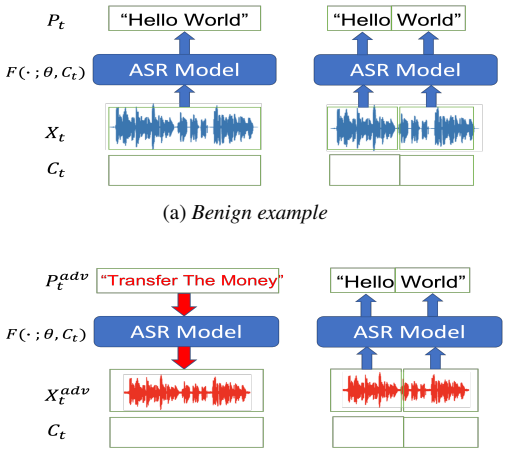
Commonly studied scenarios in adversarial attacks and defenses are based on systems that perform closed-set classification tasks. Unlike them, speech recognition is a sequence-to-sequence task where both the input and output are variable-length sequences of different units with a latent alignment between them. For this reason, the mapping from the input utterance to the output prediction is not unique although the ASR model is fixed and deterministic at inference time. More precisely, given an utterance $\mathbf{X_t}$, the ASR model can either

- take the full utterance as the input to the neural network and predict $\mathbf{Y_t}$ at once, which is true in most non-streaming end-to-end ASR models, or
- cut the utterance into successive chunks of arbitrary lengths, forward them to the network one by one, and concatenate the resulting outputs.

Formally, consider a partition function $c(\mathbf{t})$ that represents how $\mathbf{X_t}$ is cut into chunks. For instance, if we split the interval $\mathbf{t} = \{t_s, t_e\}$ into 20 ms chunks with no overlaps (i.e. use a 20ms window with a 20 ms stride), then $c(\mathbf{t}) = \{(t_s, t_s + 20), (t_s + 20, t_s + 40), ..., (t_e - 20, t_e)\}$. With $c(\mathbf{t})$ we can extend (1) as:

$$ \mathbf{P_t} = F(\mathbf{X_t}; \theta, c(\mathbf{t})) \tag{6} $$

In other words, the ASR output $F(\cdot; \theta, c(\mathbf{t}))$ depends on both the network architecture and parameters $F(\cdot; \theta)$ and the partition function $c(\mathbf{t})$.



(a) *Benign example*

(b) *Adversarial example*

Figure 2: *An illustration of chunking defense on (a) benign example and (b) adversarial example.*

3.2. Motivation

3.2.1. Lack of transferability

As discussed in Section 2, since successful targeted attacks [5, 6, 7] usually take hundreds to thousands of iterations to sucessfully perturb a benign example $\mathbf{X_t}$, the chances are high that the adversarial example $\mathbf{X_t}$ over-fits to the forward function $F(\cdot; \theta, c(\mathbf{t}))$, which depends on $c(\mathbf{t})$. This over-fitting intuition is supported by the observations of [8, 16], who report that adversarial examples do not *transfer* well from one ASR system to another, suggesting that adversarial examples in audio domain may be highly specific to the model they are generated for. As a result, if there is a mismatch between the partition functions $c(\mathbf{t})$ used to craft the adversarial examples and the one used to perform inference, we expect the attack will not succeed.

[1] https://www.darpa.mil/program/guaranteeing-ai-robustness-against-deception

Table 1: *ASR outputs of an undefended model for the same speech and different PGD attack parameters, demonstrating that mounting a successful targeted attack with a high signal-to-noise ratio (SNR) requires a large number of iterations.*

Ground Truth Transcript				there's a heavy storm coming on i cried pointing towards the horizon.
ID	ε	iters	SNR (dB)	ASR Output
(a)	0.001	7	41.94	there's a heavy storm coming on i cried twining towards the horizon.
(b)	0.01	7	22.76	where is a heavy storm coming on i cried twinning towards the horizon.
(c)	0.1	7	2.86	and queer the heavy crowing cunning are at gri when they be lucif fiery.
(d)	0.01	500	35.93	where is ahead store cutting no mother she buy drink according with she sold his furniture.
(e)	0.1	500	19.84	when she heavily get no money to buy drink with she sold his furniture.
Attacker's Target Transcript				when she could get no money to buy drink with she sold his furniture.

3.2.2. Limited receptive field

ASR systems based on modern neural network architectures have large receptive fields. This means that to predict a token at time t, the system aggregates information from several past and future time steps. For example, if the system is based on a convolutional network the receptive is proportional to the product of the kernel-sizes and number of layers; and it it is based on transformer the receptive field covers the whole utterance. Therefore, the adversarial perturbation at time t_1 can have an impact on the predicted token at time t_2.

The partition function $c(\mathbf{t})$ breaks a long utterance into multiple chunks that are processed in isolation, so that the output of one chunk is not affected by the input of another. Chunking prevents the attackers from distributing the adversarial noise needed to modify the output of a given chunk along the entire length of the utterance. Therefore, if the partition function used to generate the adversarial example (or if it is generated on the full utterance) is different from the test partition, the attack should be less effective. This is because some of the adversarial waveform samples optimized to change the $i-th$ chunk output will be at a different chunk at test time.

3.3. Defense method

Based on the discussion above, the defense is straightforward: we simply cut the utterance $\mathbf{X_t}$ into small chunks during test time, which does not require any extra work during training/fine-tuning of the ASR model, nor any other pre/post-processing modules. The utterance partition can be fixed or stochastic. The former can be effective if the attacker does not know about the defense method. However in a fully white-box scenario the attacker would break the system using the same partition to generate the attacks. On the other hand, the stochastic version will use a different partition each time we evaluate the system, making the job of the attacker more difficult. Figure 2 illustrates the expected behavior of chunking defense on benign and adversarial speech respectively. Note that, in this work we reform the chunks into a full sequence at the network output level. In other words, the decoder still receives a full sequence of output from the network instead of decoding on each chunk one by one and combining them in the end. It makes the chunking defense more general to all sequence-to-sequence tasks

4. Experimental Results

4.1. Dataset

Librispeech [17] full 960 hours corpus was used as training data for the baseline undefended ASR models. Test clean data was used to generate adversarial examples. Since generating successful targeted attacks take hundreds of PGD iterations for each example, we do all the experiments on the first 100 examples from the test clean data due to the computational limitations, which was also adopted in other works[6, 7] when generating adversarial examples on full test clean data becomes too expensive.

4.2. Attacks

As shown in Table 1, we select five different settings for PGD attack as the threat models. In all cases, $\alpha = (1.5 * \varepsilon)/I$ to let perturbation ${\sigma_\mathbf{t}}^{adv}$ reach the budget boundary ε. We performed targeted attacks following (3), target phrases Y_t^{tar} were taken from a pool of phrases in the Librispeech train set. For each test example, we chose the phrase with the closest length to the ground truth Y_t to make the attack easier and, therefore, more difficult to defend.

4.3. Non-streaming E2E LF-MMI Conformer ASR

We started with a non-streaming end-to-end model that was trained with full utterances. The system is based on the Snowfall and K2 [14] frameworks and was trained on end-to-end lattice-free MMI loss. The neural network consists of two convolution sub-sampling layers with kernel size 3*3 and stride 2, followed by a 12-layer 4-head Conformer encoder [18]. Each conformer layer uses 256 attention dimensions and 2048 feed-forward dimensions. Log Mel-Filterbank (LFB) acoustic features with 80 bins are used, which are extracted with a 25ms window, 10ms hop-length at a sample rate of 16kHz. The model was trained for 20 epochs and the parameters from last 5 checkpoints are averaged to get the final model. The conformer produces a sequence of posteriors for each input sequence, which goes to k2 for decoding next. As mentioned in Section 3.3, **the reformation of chunks into full sequences only took place at the posterior level instead of word/sentence level**.

As shown in Table 2 sys 1-5, for benign examples and weakly optimized attacks (e.g. attack (a) to (c) that only has 7 iterations), chunking defense with small chunk size doesn't help the model but instead degrades its performance in terms of increasing WER_{REF}. It is expected since the model was trained on full utterances while tested on small chunks. Such inconsistency of input scale along with the lack of context in test time makes the model perform badly, both with and without attacks. However, for highly optimized attack (e.g. (d) and (e)), chunking defense manage to break the targeted attack by increasing the WER_{TGT} dramatically.

Table 2: *Chunking defense on 2 baseline models against 5 adversarial attacks on the first 100 examples from Librispeech test clean dataset. All the attacks are generated on the full utterance in an non-adaptive way. WER_{REF} and WER_{TGT} is the WER of the output sentence w.r.t the ground truth and target sentence respectively. From the defense perspective, WER_{REF} is lower the better (↓) while WER_{TGT} is higher the better (↑).*

sys ID	Architecture	chunk size frames (secs)	WER_{REF} (%) ↓					WER_{TGT} (%) ↑	
			benign	(a)	(b)	(c)	(d)	(d)	(e)
1	E2E LFMMI	Full	4.89	22.22	61.33	90.28	64.44	87.04	42.13
2		600 (6s)	8.00	26.22	71.11	93.33	65.33	91.67	70.37
3		400 (4s)	10.22	32.00	74.67	95.56	64.89	91.67	68.98
4		300 (3s)	12.44	34.67	75.56	96.89	62.22	92.13	76.85
5		200 (2s)	28.89	41.78	75.11	95.37	69.78	93.52	86.57
6	Streaming CTC	Full	2.67	24.89	75.11	107.11	87.11	70.83	18.98
7		200 (2s)	4.44	29.78	70.22	108.00	69.33	86.57	37.50
8		64 (0.64s)	4.44	29.78	70.22	116.00	63.11	89.35	50.93
9		32 (0.32s)	5.33	29.78	72.44	111.56	63.56	88.43	55.56
10		16 (0.16s)	7.56	29.78	77.33	112.44	61.33	98.15	61.11

4.4. Streaming CTC Conformer model

To solve the degradation on benign examples, a baseline model trained with flexible chunk size is more favorable for the chunking defense. In this section, we will test the proposed defense on a streaming CTC model[2]. It has the same network architecture and feature extraction modules as the previous E2E LFMMI model. It was trained with CTC loss [19] in K2 [14] with full left context but limited random right context for streaming decoding in the test time. As shown in Table 3, the streaming model does not degrade too much when decoding benign speech in small chunks. Note that the chunk size here refers to the look-ahead right context appended to the input samples in a streaming model. For this reason, the chunk size in this model is not the same concept in the previous non-streaming model and thus not comparable among two models.

Table 3: *WER of the streaming CTC model on Librispeech test clean dataset with different chunk sizes of the right context.*

chunk size (right)	WER(%)	
	test clean	test other
Full	3.53	8.52
200 (2.00s)	3.78	9.38
64 (0.64s)	4.06	9.98
32 (0.32s)	4.30	10.55
16 (0.16s)	5.88	12.01

From Table 2 sys 6-10 we can see that, when applied to a streaming model that is capable of predicting on flexible chunks, chunking defense does negligible degradation on WER_{REF} while significantly raising the WER_{TGT} from 18.98% to 61.11%. Note that both (d) and (e) attacks take 500 iterations to optimize an example with targeted output. To get better targeted attack (e.g. $WER_{REF} \leq 5$), more iterations or distortion will be needed, in which cases chunking defense should show better performance.

4.5. Adaptive attack

To study the worst case scenario, we assume that the adversary is also going to adapt to the chunking defense by using chunks during attack instead of the full utterances.

Table 4: *WER_{TGT} (%) of the streaming CTC model attacked and defended by chunks. PGD-500-0.1 (e) is used to craft adversarial examples.*

chunk size (defense)	chunk size (attack)			
	16	32	64	200
16	**20.37**	33.8	40.74	37.5
32	31.02	**20.83**	30.09	29.63
64	50.93	32.41	**17.59**	18.52
200	50.00	43.06	37.04	**13.43**
avg	34.11	32.52	31.36	24.77

Table 4 shows that, when the partition functions $c(t)$ used to generate attacks and defend match, the adversary can find examples with low WER_{TGT} again. However, if the adversary has to use small chunks to match the defense, this prevents it from using a large receptive field of speech to better craft the example. As a result, the average WER_{TGT} increases when the chunk size decreases. It shows chunking defense can still benefit the victim model even under strong adaptive attacks, although not as significant as when using non-adaptive attacks.

5. Conclusions

In this work, we introduce the partition function to the adversarial study on sequence-to-sequence tasks and develop the chunking defense accordingly. We present its effects on two end-to-end models under five different attacks and show its full potential against targeted attack along with its limitation for weak attacks. A further experiment on adaptive attacks suggest that even when the adversary is aware of the chunking defense and generate attacks accordingly, we can still provide the model with some robustness by forcing the attacker to limit its receptive field. We look forward to applying chunking defense to more sequential tasks in the future.

6. Acknowledgements

This research was partially supported by DARPA GARD Award HR001119S0026-GARD-FP-052.

7. References

[1] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates *et al.*,

[2] https://github.com/k2-fsa/icefall/egs/librispeech/ASR/streaming_conformer_ctc

"Deep speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[2] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. CONF. IEEE Signal Processing Society, 2011.

[3] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang *et al.*, "Streaming end-to-end speech recognition for mobile devices," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6381–6385.

[4] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[5] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[6] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2018, pp. 1–7.

[7] Y. Qin, N. Carlini, G. Cottrell, I. Goodfellow, and C. Raffel, "Imperceptible, robust, and targeted adversarial examples for automatic speech recognition," in *International conference on machine learning*. PMLR, 2019, pp. 5231–5240.

[8] P. Żelasko, S. Joshi, Y. Shao, J. Villalba, J. Trmal, N. Dehak, and S. Khudanpur, "Adversarial attacks and defenses for speech recognition systems," *arXiv preprint arXiv:2103.17122*, 2021.

[9] A. Sreeram, N. Mehlman, R. Peri, D. Knox, and S. Narayanan, "Perceptual-based deep-learning denoiser as a defense against adversarial attacks on asr systems," *arXiv preprint arXiv:2107.05222*, 2021.

[10] A. Shafahi, M. Najibi, M. A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, "Adversarial training for free!" *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[11] Z. Yang, B. Li, P.-Y. Chen, and D. Song, "Characterizing audio adversarial examples using temporal dependency," *arXiv preprint arXiv:1809.10875*, 2018.

[12] H. Hadian, H. Sameti, D. Povey, and S. Khudanpur, "End-to-end speech recognition using lattice-free mmi." in *Interspeech*, 2018, pp. 12–16.

[13] Y. Shao, Y. Wang, D. Povey, and S. Khudanpur, "Pychain: A fully parallelized pytorch implementation of lf-mmi for end-to-end asr," *arXiv preprint arXiv:2005.09824*, 2020.

[14] D. Povey et al., "k2," [Online] Available: https://github.com/k2-fsa/k2.

[15] R. Olivier and B. Raj, "Sequential randomized smoothing for adversarially robust speech recognition," *arXiv preprint arXiv:2112.03000*, 2021.

[16] H. Abdullah, K. Warren, V. Bindschaedler, N. Papernot, and P. Traynor, "Sok: The faults in our asrs: An overview of attacks against automatic speech recognition and speaker identification systems," in *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2021, pp. 730–747.

[17] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[18] 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," *arXiv preprint arXiv:2005.08100*, 2020.

[19] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.