INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

SCADA: Stochastic, Consistent and Adversarial Data Augmentation to Improve ASR

Gary Wang[1], *Andrew Rosenberg*[2], *Zhehuai Chen*[2], *Yu Zhang*[2], *Bhuvana Ramabhadran*[2], *Pedro Moreno*[2]

[1]Simon Fraser University
[2]Google

ywa289@sfu.ca, {rosenberg,zhehuai,ngyuzh,bhuv,pedro}@google.com

Abstract

Recent developments in data augmentation has brought great gains in improvement for automatic speech recognition (ASR). Parallel developments in augmentation policy search in computer vision domain has shown improvements in model performance and robustness. In addition, recent developments in semi-supervised learning has shown that consistency measures are crucial for performance and robustness. In this work, we demonstrate that combining augmentation policies with consistency measures and model regularization can greatly improve speech recognition performance. Using the Librispeech task, we show: 1) symmetric consistency measures such as the Jensen-Shannon Divergence provide 4% relative improvements in ASR performance; 2) Augmented adversarial inputs using Virtual Adversarial Noise (VAT) provides 12% relative win; and 3) random sampling from arbitrary combination of augmentation policies yields the best policy. These contributions result in an overall reduction in Word Error Rate (WER) of 15% relative on the Librispeech task presented in this paper.

Index Terms: speech recognition, CoDA, Augmentation Policy, RandAug, Jensen-Shannon

1. Introduction

Data augmentation is widely used for creating additional training data for machine learning systems, ranging from applications in computer vision[1, 2] to speech recognition[3, 4, 5, **?**, **?**]. Recent work with deep learning systems have show that data augmentation can also greatly improve accuracy [6], robustness [7, 8] and deliver substantial improvements to semi-supervised learning framework[9, 10, 11]. A natural extension to naive application of data augmentation is in the development and search for optimal augmentation policies[6, 12, 13, 14]. This line of work focuses on searching (either directly or through some proxy task) for a set of augmentation schedules and sub-parameters that gives great model performance across datasets and model configurations. A contrastive learning framework to decide on the choice of augmentations, with an additional learnable nonlinear transformation between the augmented representations and the contrastive loss was introduced in [15]. This work on ImageNet, highlighted the importance of data augmentation schemes for unsupervised/semi-supervised learning methods. The empirical results from this line of work shows that optimal augmentation policies can achieve better gains than directly applying data augmentation methods. In automatic speech recognition (ASR), recent augmentation methods such as SpecAugment [16] have provided gains in performance across a wide range of datasets and models. However, there has not been much work in the literature on combining these types of regularization and data augmentations schemes such as multistyle training. In this work, we extend upon SpecAugment and other augmentation techniques to see how they can be composed and combined to give greater improvements in performance for ASR.

This work also expands upon techniques from previous work [17] that applies consistency regularization when incorporating Text-to-Speech (TTS) with ASR. In [17], consistent data augmentation (CoDA) was applied to ensure ASR predictions are consistent when presented with real and TTS synthesized utterances. CoDA loss was crucial to make ASR model domain agnostic, resulting in improved performance. In this work, we explore a range of alternative consistency measures to stabilize training and generalization based on augmented data especially when augmentations are drawn from a diverse population of techniques. In addition, we investigate a closely related model regularization technique called Virtual Adversarial Training[18] that adds adversarial noise on top of our augmented model inputs, and show that it provides complementary gains for model performance. While there have been other applications of VAT on speech tasks (e.g. [19, 20]), we believe this is the first use of VAT in training a sequence model as in ASR.

The contributions of this paper are as follows:

- Novel application on combining augmentation policies with consistency regularization on supervised training for ASR
- We evaluate how to best compose different augmentation policies
- Assessment of a variety of consistency measures help improve model performance
- Demonstration of the value of adversarial model regularization technique such as Virtual Adversarial Training complement augmentation policies and consistency regularization

2. Related Work

2.1. Augmentation Strategies

In this Section, we present recent work on data augmentation policy search. It should be noted that much of the augmentation policy search work has been centered around ImageNet. In AutoAugment[6], a reinforcement learning (RL) controller was trained to select the best augmentation policies from performance metrics derived from a small proxy task. Fast AutoAugment[12] replaces the RL policy search with a method that directly searches for augmentation policies that maximize the match between the distribution of augmented split

http://dx.doi.org/10.21437/Interspeech.2020-2920

and the distribution of another, unaugmented split, thus speeding up policy search. In population based augmentation[13], the augmentation policy search was replaced with population based[21] policy search. RandAugment[14] improves upon these methods by removing the need to search over the small proxy, by instead uniform random sampling between a set of strong and diverse augmentations. The results of RandAugment suggest that the assuming that the set of augmentation candidates are similarly useful, the gains from learning a specific policy are negligible compared to a random selection policy.

2.2. Consistency Regularization

Consistency regularization has been utilized effectively in semi-supervised learning framework ([22, 23, 10]). Various consistency measures enforce model predictions to be robust to small perturbations in input. The motivation behind consistency regularization is that small changes in the input should lead to small changes in the output. This is then expanded to suggest that a robust model should behave similarly in response to original and augmented features.

In Unsupervised Data Augmentation (UDA) [10], unpaired data x are passed through various augmentation methods to obtain augmented unpaired data $\hat{x}$. A KL-Divergence consistency loss was then applied on the respective model predictions $p_\theta(y|x)$ and $p_\theta(y|\hat{x})$ to ensure model predictions are consistent.

In FixMatch [11], consistency regularization and pseudo-labelling were combined to improve upon previous semi-supervised learning works. Weak augmentation was applied to unpaired data to generate model predictions, which act as the pseudo-label after sharpening. Strong augmentations such as RandAugment was then applied and subsequent model predictions are obtained and used to train via cross entropy loss on labels generated from weak augmentations.

The above mentioned methods have been mostly applied within a semi-supervised learning setup, where consistency measures were applied on unpaired data. AugMix [8] is one of few works that applies consistency measures on supervised learning, by applying Jensen-Shannon Divergence loss along with mixing augmentation policy. This work shares similarities to AugMix in that we also apply augmentation policies and consistency regularization in the fully supervised training domain for ASR.

A closely related work in semi-supervised learning and model regularization is Virtual Adversarial Training [18] (VAT). In this work, model distributional smoothness was defined for a given input data as the KL-divergence based robustness of the model against local, adversarial perturbations around said data point. VAT can be viewed from a model regularization perspective in that it helps models be robust to adversarial noise. It can also be viewed as an additional data augmentation strategy in that it adds local adversarial noise to model inputs without requirement of labels.

3. SCADA: Stochastic, Consistent and Adversarial Data Augmentation

In this section we describe model architectures, training details and how we apply augmentation policies, and consistency measures including specifics of VAT training.

3.1. Model and Training Details

For all ASR experiments, we use listen-attend-spell (LAS) model[24]. Specifically, the LAS encoder consists of 2 convolutional layers of 32 filters with shape 3×1 and 2×2 stride, followed by four bidirectional LSTM layers of 1024 units for each direction. The LAS decoder consists of locally sensitive attention followed by two unidirectional LSTM layers with 1024 units.

We use 80 dimension melspectrogram features with delta and double-delta stacking. Targets are 16k Word Piece model (WPM) subword vocab. Decoding is performed without any second-pass rescoring. Training is performed using Adam[25] for 200k steps. We use a warmup and exponential decay learning rate schedule with a maximum of 1e-3 and minimum of 1e-5.

3.2. Augmentation Methods

In this section we describe the augmentation methods used within these experiments. These augmentations all are applied on mel filterbank outputs rather than the waveform directly.

SpecAugment SpecAugment [16] is one of our core augmentation methods. SpecAugment randomly applies time and frequency masks to input melspectrograms. We use 2 SpecAugment configurations in our experiments (*SP1* and *SP2*), both providing similar performance individually. *SP1* uses 4 time masks with a max masking ratio of 0.1, and 1 frequency mask with a max of 15 bins. *SP2* uses 6 time masks with a max masking ratio of 0.1, and 3 frequency masks with a max of 15 bins. The size of the masked time and frequency bands are uniformly sampled between 0 and their max value. Note, while [16] describes SpecAugment as including a time warping augmentation, we don't find any performance improvements from including this, so omit it.

Low Pass Smoothing We apply low pass smoothing as augmentation on melspectrogram. A 2-D isotropic gaussian kernel is used, with mean value of 0 and uniform sampled standard deviation in the range of $[0, 0.2]$. The 2-D guassian kernel is then convolved with melspectrogram with kernel of shape 5×5. Note that the stochasticity is introduced by sampling standard deviation values instead of using a fixed constant.

Additive Scaled Guassian Noise We apply additive guassian noise as augmentation on melspectrogram. We scale the gaussian noise addition to the mean of melspectrogram values with a specified noise-to-signal ratio (NSR) of 0.2. Stochasticity is added by uniform sampling the NSR ratio in the range of $[0, 0.2]$.

3.3. Stochastic Augmentation Policies

RandAugment Policy Following [14], we create a stochastic RandAugment policy $RA(p_1, p_2..p_k)$ that will uniformly sample between k different augmentations provided during every call. We implement RandAugment as another augmentation method as to allow for arbitrary composition and combination of different augmentation policies. As an example, we can compose two distinct RandAugment policies inside another RandAugment policy:

$$\begin{aligned} raug_1 &= RA(p_1, p_2, p_3) \\ raug_2 &= RA(p_4, p_5) \\ raug_3 &= RA(raug_1, raug_2) \end{aligned} \quad (1)$$

Augmentation Stacking Inspired by AugMix [8], we apply augmentation stacking in order to get more diverse augmentations with every frop call. Specifically, we apply augmentation in a two stage stacking process.

In the first stage, we create A RandAugment policy **RA-pre** that randomly samples between three distinct augmentations: 1) identity, 2): low pass and 3: scaled guassian noise. In the second stage, we create a RandAugment policy $RA_{specaug}$ that randomly samples between 2 distinct and equally performant SpecAugment configurations.

This construction allows us to stack these augmentations in a sequential fashion. Given an input mel spectrogram x, we first apply **RA-pre**, followed by **RA-spec** to obtain $\hat{x}$.

3.4. Loss Definitions

For each data point (x, y), two separate data augmentations are applied to the same batch of input data to obtain x_1 and x_2. Given these pairs of inputs, we apply our model M and obtain model hypothesises $p_\theta(y|x_1)$ and $p_\theta(y|x_2)$. We apply regular cross entropy training with ground-truth labels y, and obtain our supervised loss:

$$\mathcal{J}_{sup} = \mathbb{E}_{x_1, y \in L} p_\theta(y|x_1) + \mathbb{E}_{x_2, y \in L} p_\theta(y|x_2) \tag{2}$$

We also calculate consistency loss on the pair of model predictions, $L_{consist}(y_1^*, y_2^*)$, where y_1^* and y_2^* are model hypothesises with two different augmentation variants. Note that y^* here can represent both ASR model encoder states as well as decoder logits and include this as a regularization term on the final loss.

3.5. Consistency Measures

In this section we describe the consistency measures applied both on the ASR encoder and decoder side between two distinct augmentations.

Encoder Consistency Encoder consistency ensures that ASR encoder states are domain agnostic when seeing different augmented inputs. Given two distinct augmented inputs x_1 and x_2, we obtain ASR encoder oupututs as e_1 and e_2. We then define encoder consistency loss:

$$\mathcal{J}_{enc} = ||e_1 - e_2||^2 \tag{3}$$

Consistency Loss In previous work [17], we applied Consistency Data Augmentation (CoDA) to ensure consistent ASR predictions between ground-truth and TTS synthesized utterances. In this work, both features are augmented versions of a source mel-spectrogram. Since the augmentation does not rely on the label y to modify x, this loss is equivalent to that used in UDA [10]:

$$\mathcal{J}_{uda} = \mathbb{E}_{x \in U} \mathbb{E}_{\hat{x} \sim q(\hat{x}|x)} \mathcal{D}_{KL}(p_\theta(y|x) || p_\theta(y|\hat{x})) \tag{4}$$

Despite KL divergence being asymmetric, it proved useful in our previous work as it pushed TTS augmented ASR decoder logits closer to ground-truth. In this work however, all augmentations are equal, and equal weight should be applied to all distinct augmentations.

Jensen Shannon Divergence To enforce smoother network response and have a symmetric loss, we utilize the Jensen Shannon Divergence (JS) between augmentations. Specifically, given two augmented inputs x_1 and x_2 and their respective model prediction posterior distributions $p_1 = p_\theta(y|x_1)$ and $p_2 = p_\theta(y|x_2)$, we compute JS loss as:

$$\begin{aligned} M &= \frac{1}{2}(p_1 + p_2) \\ JS(p_1, p_2) &= \frac{1}{2}(KL[p_1||M] + KL[p_2||M]) \end{aligned} \tag{5}$$

Virtual Adversarial Noise Virtual Adversarial Training (VAT) [18], computes adversarial noise r_{adv} on an input x to obtain an **adversarially** augmented input $\hat{x}$. r_{adv} is selected to be a small change in the direction that would increase the loss of the model most. The magnitude of the adversarial noise is a hyperparameter. We set the VAT vector norm length to 10.0 for all experiments if not otherwise specified. The VAT regularization loss is the divergence between the predictions

$$\mathcal{J}_{vat} = \mathcal{D}_{KL}\left(p(y|x) || p(y|x + r_{adv})\right). \tag{6}$$

To apply VAT to sequential data, r_{adv} is a tensor of the same size as x. The gradient is calculated with respect to teacher-forced training to the target y to provide stability to the gradient used in the generation of r_{adv}.

4. Results and Discussion

In this section we evaluate aspects of stochastic data augmentation policy selection (Section 4.1), consistency measures (Section 4.2) and VAT regularization (Section 4.3). Finally, we combine the most effective techniques (Section 4.4.)

4.1. Augmentation Policy Comparison

We compare a variety of stochastic augmentation policies.

Since SpecAugment provides substantial and consistent improvements, our baseline includes *SP1* SpecAugment.

RA-Pre is RandAugment selecting between identity, Low Pass and Scaled Gaussian Noise with equal rates. Prior to SpecAugment.

RA-Spec describes a RandAugment process selecting from *SP1* and *SP2*. This replaces the use of *SP1* in the baseline model.

Table 1: *Stochastic Augmentation Comparison*

Description	test-clean	test-other
Baseline	4.70	15.40
Base + RA-Spec	4.60	13.75
Baseline + RA-Pre + RA-Spec	4.66	13.60

We find that stochastic augmentation policies can outperform the static parameterization of SpecAugment used in our baseline model. We observe improvements from stochastically manipulating the magnitude of SpecAugment parameters. This is particularly striking in the improvement to test-other. SpecAugment (and RA-Spec) completely mask out time and frequency bands from the input signal. When we combine this with more traditional augmentation techniques as in **RA-Pre**, we find some additional gains to test-other, but a modest regression on test-clean.

4.2. Importance of Consistency Measures

Here we compare consistency measures applied to the encoder and decoder of the ASR model. Overall, we find that applying

Table 2: *Consistency Loss Comparison*

Description	test-clean	test-other
Baseline	4.7	15.4
Baseline + RA-Pre + 2 Forward	4.8	15.0
Baseline + RA-Pre + Encoder L2	6.2	18.0
Baseline + RA-Pre + Decoder JS	5.0	14.8

Table 3: *Regularization with VAT*

Description	test-clean	test-other
Baseline	5.9	17.7
Baseline + VAT (norm=6)	5.9	16.9
Baseline + VAT (norm=10)	5.3	16.8

a consistency loss between two different manipulations using SpecAugment leads to substantial WER reduction.

Results of these experiments can be found in Table 2, each consistency measure is applied after RA-Pre with the *SP1* SpecAugment configuration on each input. We find that measuring consistency of the model's outputs (i.e. decoder outputs) to be more reliable than consistency in the encoder output. We were unable to find a configuration that yielded any improvement by regularizing the consistency in encoder outputs. One explanation for this is that the semantics of the decoder are more clearly understood than those of the encoder. The decoder outputs are WPM units. Measuring differences in this output space is well understood, and optimized through cross-entropy here. (Though, of course, other distance measures have are well motivated and used to optimize WER performance e.g. minimize word error rate, etc.) On the other hand, it is not clear how perturbations in the encoder output lead to different model behaviors. It is likely that the semantics of the encoder output space is not well measured by L2 distance.

When calculating consistency loss, we make two forward passes through the model with different augmented features. While equivalent in the limit, the baseline model will see half as many augmentations in the same number of backprop steps as the models trained with consistency losses. The *Baseline + 2 Forward* experiment trains with two forward passes, accumulating losses from two augmentations of the same data point *without* applying a consistency loss term. This lets us measure the impact of the duplicated forward pass (and making updates supported by multiple augmentations of the same data point) from the value of consistency measures. We find a small improvement from this duplicated forward pass training, but consistency regularization by Jensen-Shannon (JS) divergence provides a decent consistency signal. In [17], we used KL divergence when comparing consistency between clean speech and a synthesized version of the same utterance. In that case there was a clearly superior signal; the model should match the synthesized output toward the clean output. In this case, neither augmentation copy is a natural target for the update, making JS divergence a more appropriate regularizer. We find JSD regularization to lead to a generally more robust model, generalizing from clean training data in LibriSpeech-460 to the test-other evaluation data. However, this improvement brings with it lower performance on the in-domain test-clean evaluation set.

4.3. Tuning VAT Regularization

In Table 3, we report results from including VAT on model training. Since VAT results require two forward passes for each update step, experiments are much slower. Therefore we tune VAT parameters on with a small training budget, using only 24k training steps. Experiments using VAT with a larger training budget are reported in Section 4.4. We find VAT to provide a substantial improvement to model training over the baseline. One attractive property of VAT is that there is only a single hyperparameter, namely a norm for the size of the adversarial step. For this model, we find a larger norm to be effective.

4.4. SCADA Experiments

In the final set of experiments, we investigate how the most effective approaches to Stochastic, Consistent and Adversarial Data Augmentation (SCADA) training operates in aggregate. Table 4 contains results of these experiments. In each of these experiments, we use stochastic data augmentation policy selection incorporating both RA-Pre and RA-Spec. We find this component is effective in improving performance on test-other, while not degrading clean results. Including two alternative augmentation paths and using Jensen-Shannon consistency (JS) delivers similar improvement to test-other, but shows a degradation of clean performance. On the other hand VAT, a consistency regularization based on an adversarially constructed "augmentation", delivers additional improvement to test-clean. When we use both VAT and JSD, we obtain our best test-other performance. However, this configuration sacrifices performance on the test-clean set, a finding we observed by JSD in Table 2 as well.

Table 4: *Top line experiment*

Description	test-clean	test-other
Baseline (B)	4.7	15.4
B + RA-Pre + RA-Spec	4.7	13.6
B + RA-Pre + RA-Spec + JSD	4.9	13.4
B + RA-Pre + RA-Spec + VAT	**4.5**	13.4
B + RA-Pre + RA-Spec + JSD + VAT	4.9	**13.2**

5. Conclusion

We have shown that data augmentation techniques can be combined with consistency regularization to yield significant performance wins on supervised learning for speech recognition tasks particularly when generalizing to noisier speech. Using the LibriSpeech task as an example, we derive the following messages: 1) symmetric consistency measures such as the Jensen-Shannon Divergence provide 4% relative improvements in ASR performance; 2) Augmented adversarial inputs using Virtual Adversarial Noise (VAT) provides 12% relative win; and 3) Random sampling from arbitrary combination of augmentation policies yields the best policy. These contributions result in an overall reduction in Word Error Rate (WER) of 15% relative on the Librispeech test-other set training with 460 hours of data. The combination of these three strategies results in a WER close to an ASR system training with almost double the amount of data (960 hours) demonstrating the potential of the approach presented in this paper.

6. References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[2] T. DeVries and G. W. Taylor, "Dataset augmentation in feature space," *arXiv preprint arXiv:1702.05538*, 2017.

[3] N. Kanda, R. Takeda, and Y. Obuchi, "Elastic spectral distortion for low resource speech recognition with deep neural networks," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE, 2013, pp. 309–314.

[4] 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.

[5] 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.

[6] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation strategies from data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 113–123.

[7] R. G. Lopes, D. Yin, B. Poole, J. Gilmer, and E. D. Cubuk, "Improving robustness without sacrificing accuracy with patch gaussian augmentation," *arXiv preprint arXiv:1906.02611*, 2019.

[8] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan, "Augmix: A simple data processing method to improve robustness and uncertainty," *arXiv preprint arXiv:1912.02781*, 2019.

[9] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," in *Advances in Neural Information Processing Systems*, 2019, pp. 5050–5060.

[10] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le, "Unsupervised data augmentation," *arXiv preprint arXiv:1904.12848*, 2019.

[11] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *arXiv preprint arXiv:2001.07685*, 2020.

[12] S. Lim, I. Kim, T. Kim, C. Kim, and S. Kim, "Fast autoaugment," in *Advances in Neural Information Processing Systems*, 2019, pp. 6662–6672.

[13] D. Ho, E. Liang, I. Stoica, P. Abbeel, and X. Chen, "Population based augmentation: Efficient learning of augmentation policy schedules," *arXiv preprint arXiv:1905.05393*, 2019.

[14] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical data augmentation with no separate search," *arXiv preprint arXiv:1909.13719*, 2019.

[15] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," *arXiv preprint arXiv:2002.05709*, 2020.

[16] 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.

[17] G. Wang, A. Rosenberg, Z. Chen, Y. Zhang, B. Ramabhadran, Y. Wu, and P. Moreno, "Improving speech recognition using consistent predictions on synthesized speech," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7029–7033.

[18] 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.

[19] X. Wang, S. Sun, and L. Xie, "Virtual adversarial training for ds-cnn based small-footprint keyword spotting," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 607–612.

[20] M. Zöhrer and F. Pernkopf, "Virtual adversarial training and data augmentation for acoustic event detection with gated recurrent neural networks." in *Interspeech*, 2017, pp. 493–497.

[21] M. Jaderberg, V. Dalibard, S. Osindero, W. M. Czarnecki, J. Donahue, A. Razavi, O. Vinyals, T. Green, I. Dunning, K. Simonyan *et al.*, "Population based training of neural networks," *arXiv preprint arXiv:1711.09846*, 2017.

[22] P. Bachman, O. Alsharif, and D. Precup, "Learning with pseudo-ensembles," in *Advances in neural information processing systems*, 2014, pp. 3365–3373.

[23] M. Sajjadi, M. Javanmardi, and T. Tasdizen, "Regularization with stochastic transformations and perturbations for deep semi-supervised learning," in *Advances in neural information processing systems*, 2016, pp. 1163–1171.

[24] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.

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