Interspeech 2024
1-5 September 2024, Kos, Greece

ROAR: Reinforcing Original to Augmented Data Ratio Dynamics for Wav2Vec2.0 Based ASR

Vishwanath Pratap Singh[1], *Federico Malato*[1], *Ville Hautamäki*[1], *Md. Sahidullah*[2,3], *Tomi Kinnunen*[1]

[1]University of Eastern Finland, Finland
[2]Institute for Advancing Intelligence, TCG CREST, India
[3]Academy of Scientific and Innovative Research (AcSIR), India
{vsingh, federico.malato,tomi.kinnunen}@uef.fi,{ville,sahidullahmd}@gmail.com

Abstract

While automatic speech recognition (ASR) greatly benefits from data augmentation, the augmentation recipes themselves tend to be heuristic. In this paper, we address one of the heuristic approach associated with balancing the right amount of augmented data in ASR training by introducing a reinforcement learning (RL) based dynamic adjustment of original-to-augmented data ratio (OAR). Unlike the fixed OAR approach in conventional data augmentation, our proposed method employs a deep Q-network (DQN) as the RL mechanism to learn the optimal dynamics of OAR throughout the wav2vec2.0 based ASR training. We conduct experiments using the LibriSpeech dataset with varying amounts of training data, specifically, the 10Min, 1H, 10H, and 100H splits to evaluate the efficacy of the proposed method under different data conditions. Our proposed method, on average, achieves a relative improvement of 4.96% over the open-source wav2vec2.0 base model on standard LibriSpeech test sets.

Index Terms: speech recognition, reinforcement learning, data augmentation, wav2vec2.0

1. Introduction

Data augmentation has emerged as a common strategy for model generalization and for increasing the quantity of training data to train the *automatic speech recognition* (ASR) systems [1]. Beyond merely increasing quantity, data augmentation also introduces diversity into the training dataset, thereby reducing the risk of overfitting [2, 3]. Consequently, data augmentation has been extensively used for improving ASR systems in application-agnostic [4], low-resource [5, 6], multi-lingual [7], and children's [8] speech recognition scenarios.

While the advantages of data augmentation are apparent, a methodological challenge lies in selecting the optimal data augmentation methods, including their hyperparameters (such as signal-to-noise ratio (SNR) or speed modification factor), order (e.g., noise followed by speed modification, or noise followed by room impulse response (RIR)), and the volume of augmented data. In most of the previous studies [4, 9], the authors typically rely on heuristic ideas for choosing the augmentation methods, associated hyperparameters, and the amount of augmented data. Those are not necessarily optimum for different datasets and tasks; and choosing the right configuration remains an open challenge [10].

In this paper, we tackle the aforementioned challenge by exploring automatic methods for setting specific data augmentation hyperparameters during training. In particular, we address automatic adjustment of the proportion by which the original and augmented data get mixed during different stages of training. For easier reference, we denote this quantity as the *original-to-augmented data ratio* (OAR). Previous literature [4, 11] highlights that exceeding the amount of augmented data from certain methods beyond a certain limit can lead to a degradation in ASR performance. On the other hand, if the augmented data ratio is kept too small, it might not lead to improvements because the limited input size may not significantly impact the ASR learning process. Thus, balancing this ratio emerges as a crucial factor in training robust and accurate ASR systems.

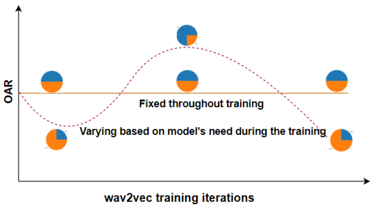

OAR
Fixed throughout training
Varying based on model's need during the training
wav2vec training iterations

Figure 1: *Two possible scenarios on dynamics of original-to-augmented data ratio (OAR) throughout the wav2vec2.0 training. The solid line (along with the fixed proportion in pie-chart) indicates fixed OAR used throughout the training. The dashed line (along with varying proportions in pie-chart) indicates our proposal, where OAR is allowed to vary dynamically throughout training.*

Another fundamental question concerns the *temporal dynamics* of the OAR throughout training. As illustrated in Fig. 1, should this ratio remain constant—or would model training benefit from dynamic adjustments? We assume so, simply because the model's behavior evolves over the course of training. At the outset, when the model has not yet encountered substantial amounts of data, a lower OAR might be preferable, so as to better initialization of model parameters at the beginning of the training. However, as the model matures and starts adapting to the training set, a higher ratio may be beneficial for avoiding overfitting. In this paper, we hypothesize through experimentation that the dynamic adjustment of OAR is indeed beneficial in model training.

Precisely, to address these concerns associated with dynamic adjustment of OAR, we propose **ROAR**, a novel approach that leverages *reinforcement learning* (RL), so-called the specific RL approach, *deep Q-network* (DQN) [12]. The choice of RL as our workhorse is rooted in its capacity to address the intricate decision-making dynamics. We also chose DQN for its sample efficiency and ensured convergence properties [12]. Sample efficiency is an important consideration as often ASR models are trained for a fixed number of iterations, which limits the number of samples available from the wav2vec2.0 training environment for DQN training. Moreover, our ASR training environment dynamics benefit from exploration strategies such as ϵ-greedy [13]. Our main contribution lies in proposing the DQN-based dynamic adjustment of OAR throughout the ASR training based on the ASR model's need for augmented data.

10.21437/Interspeech.2024-1403

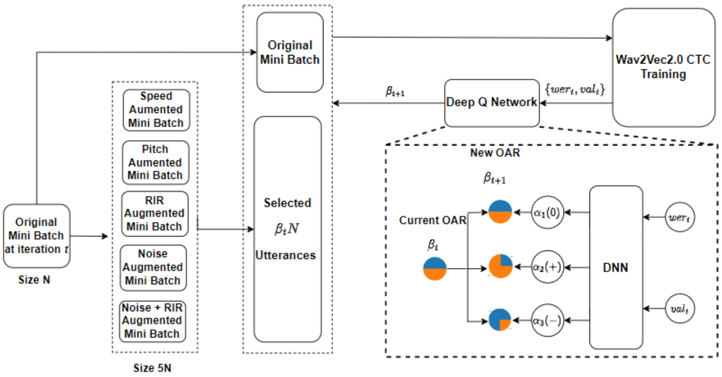


Figure 2: *Proposed ROAR based wav2vec2.0 CTC training pipeline, and illustration of Deep Q-network. β_t indicates the original to augmented data ratio (OAR) at t^{th} iteration of training, val_t and wer_t is validation loss and validation WER after t^{th} iteration of wav2vec2.0 CTC training. $\alpha_1(0)$, $\alpha_2(+)$, and $\alpha_3(-)$ indicate the actions corresponding to no change (i.e. null action), increase, and decrease in OAR, respectively.*

2. Related Work

Previously, automatic learning of augmentation policies using population based training (PBT) has been explored for automatic speech recognition (ASR) tasks [14]. However, to the best of our knowledge, RL-aided data augmentation has not been explored in the speech processing domain but considerable research has been conducted in the field of image processing [15, 16, 17] and natural language processing (NLP) [18]. Specifically, authors in [15, 16] model the 3-dimensional augmentation parameters namely, the magnitude by which the augmentation is to be applied (such as *signal-to-noise ratio* (SNR)), the augmentation policy such as rotation followed by translation, or the translation followed by rotation, and augmentation probability.

However, the idea introduced by us in this paper is different from those studied in the above image-processing and NLP domain. Firstly, we focus entirely on modeling the amount of augmented data through the OAR (β) as shown in Fig. 2. Secondly, our RL training approach is different from those proposed in [15, 16]. Specifically, authors in [15] use the neural architecture search with RL [19]. On the other hand, we use a DQN-based RL strategy by deriving the reward for every K^{th} iteration during the wav2vec2.0 CTC fine-tuning [20] which reduces the training time substantially. Thirdly, the motivation behind the augmentation in image processing is to create visual diversity by rotation, translation, and blurring of original images. Similarly, diversifying contextual diversity of particular words remains the key to NLP augmentations. While, in the case of speech, the augmentation methods revolve around the speaker, speaking style, and surrounding diversity. Hence, exploring the RL-aided augmentation for speech itself is one of the novel aspects of this paper.

3. Proposed ROAR Method

3.1. Background on DQN

Deep Q-networks (DQN) [12], developed as a combination of *Q-learning* [21] and deep neural networks [22], is designed to handle high-dimensional state spaces by approximating the optimal action-value function. The core idea involves training a neural network to predict the Q-values for each possible action in a given state, enabling the agent to make informed decisions. Q-values signify the predicted cumulative reward that an agent expects to obtain by taking a specific action in a particular state. DQN agent interacts with an environment through a sequence of observations, actions, and rewards while storing the past experiences in a replay buffer. Past records are then used to update the policy, increasing the probability of selecting an action that maximizes the reward. DNN approximated Q value is modeled using a state value function given by:

$$Q^*(s,a) = \max_{\pi} \mathbb{E}\left[\sum_{\tau=1}^{T} \gamma^{\tau} r_{t+\tau} | s_t = s, a_t = a\right] \quad (1)$$

where $Q^*(s,a)$ is the DNN approximated Q-value for state s, and action a, achievable by a policy $\pi = P(a|s)$, r_t is the reward at step t, γ is the discount factor for future rewards, T is the time horizon over which the sum is computed.

During the training phase, the DQN agent systematically navigates the environment through multiple iterations, commencing exploration from the initial states and persisting until a terminal state or predefined horizon is attained. These iterative traversals are commonly denoted as training episodes. Across these episodes, the DQN agent consistently refines its policy, aiming for optimal decision-making.

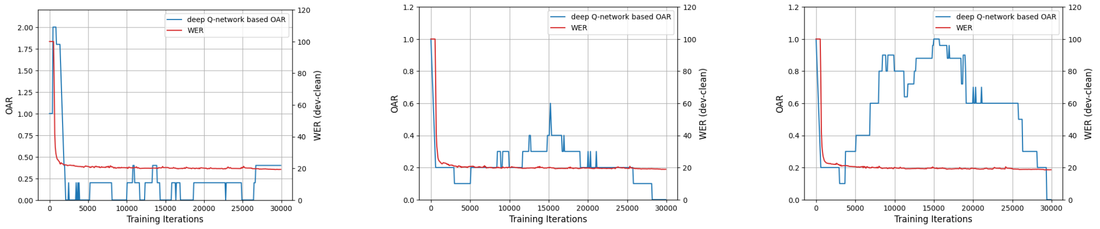
3.2. Using DQN to Adjust OAR

Our proposed method, *ROAR*, models the wav2vec2.0 CTC training [20] environment using a 2-dimensional space comprising validation loss and validation WER. We model the action space as a 3-dimensional, discrete space regulating the value of the current original to augmented data ratio. We set action 1 to be the null action, while actions 2 and 3 correspond to an increase and decrease of 0.2 of the current value, respectively. At each timestep, our DQN agent greedily selects the best action. Then, new batches of augmented data are generated and used for training wav2vec2.0 CTC model for a fixed number of iterations. Finally, we generate the reward from the change in WER. Hence, our proposed method benefits from the non-differentiable objective such as WER. Further, this approach offers a nuanced perspective, allowing the model to adapt to its evolving learning needs, ultimately enhancing the performance and robustness of wav2vec2.0 [20] based ASR system. We show our DQN training setup and the overall wav2vec2.0 training pipeline in Fig. 2.

4. Experimental Setup

4.1. Dataset

We utilize the Librispeech corpus [23] to validate our proposed ROAR method. In particular, we use the same four training subsets as in [20] consisting of, respectively, 10 minutes, 1 hour, 10 hours, and 100 hours of data. We evaluate the baselines and proposed models on standard `dev-clean`, `dev-other`, `test-clean`, and `test-other` evaluation sets. `Dev-clean` is used as a validation set for deriving the reward for RL-based training pipeline, and also selecting the best (in terms of validation WER) baselines and proposed model checkpoints.



(a) *Episode-1: OAR dynamics*

(b) *Episode-2: OAR dynamics*

(c) *Episode-3: OAR dynamics*

Figure 3: *Depiction Deep Q-network based original-to-augmented data ratio (OAR) dynamics throughout the wav2vec2.0 training on Librispeech 1H training split and visualization of the evolution OAR across various episodes.*

4.2. ASR System

We consider the frequently used open-source wav2vec2.0 [20] based ASR system in our experiments. The wav2vec2.0 is available in two configurations, namely, *Base* and *Large*. We utilize the Base configuration for our experimentation which includes 12 transformer [24] blocks, a model dimension of 768, an inner dimension of 3,072, and 8 attention heads, totaling 94 million parameters.

The wav2vec2.0 training comprises two stages. The initial stage involves pretraining, which is a contrastive loss [25] based unsupervised training aimed at generating application-independent audio embeddings. Checkpoints available from this stage of training are referred to as *self-supervised* checkpoints. In the second stage, referred to as the fine-tuning phase, an output layer is added, and the entire model is fine-tuned with CTC [26] loss. Checkpoints available from this stage of training are referred to as *pre-trained* ASR. All publicly available wav2vec2.0 [20] self-supervised checkpoints and pre-trained ASR are trained without augmentation.

In our experiments, we forego the first stage training and instead utilize open-source self-supervised wav2vec2.0 base checkpoint [1] and finetune them using CTC loss [26]. We explore two scenarios in the second stage CTC training: in the first scenario, we randomly initialize the output layer on top of the self-supervised wav2vec2.0 checkpoint and train the model under different data conditions. In the second scenario, we utilize the open-source pre-trained ASR wav2vec2.0 model [2], already trained on 100 Hours LibriSpeech using CTC loss, and further train it on augmented data.

The rationale behind the second scenario is rooted in the working mechanism of DQN. Our DQN derives reward from the change in WER, which is expected to decrease substantially at the beginning of training in the first scenario, as the weights are randomly initialized and WER is at its max. Hence, the reward is always positive for DQN at the beginning, even if it makes a few wrong decisions. On the other hand, in the second scenario, the open-source fine-tuned wav2vec2.0 is already optimal and only the right OAR (i.e. right DQN action) will lead to the decrease in WER. This ensures the reinforcement of DQN from the beginning of the training.

4.3. Baseline

Our baseline ASR systems include state-of-the-art wav2vec2.0 trained with CTC loss for different but fixed original to augmented data (OAR). OAR 0 indicates the standard wav2vec2.0 CTC training where no augmentation is used. For reference, we also include the results with the open-source wav2vec2.0 model (wherever available).

4.4. Data Augmentation Methods

In this study, we employ the following commonly used five data augmentation techniques to generate augmented data in experimentation:

Noise: The addition of background noise to the audio data with signal-to-noise (SNR) sampled uniformly between 0–20 dB, simulating real-world conditions [27]. ***RIR***: Room impulse response-based augmentations that emulate different acoustic environments [28]. ***Noise and RIR***: First Noise and then RIR are applied simultaneously on time domain speech signal. ***SM***: Speed modification with factor sampled uniformly between $0.9 - 1.1$ [1]. ***PM***: Pitch modification with factor sampled uniformly between $0.9 - 1.1$ [1].

4.5. Deep Q-Network

Our DQN model configuration includes the most commonly used *epsilon-greedy* Q-policy [21] with a learning rate set to 0.001, a discount factor γ of 0.99, warm-up steps 50, a replay buffer size of 10,000, and a batch size of 32 for experience replay. The neural network architecture consisted of two fully connected layers with 64 neurons each, employing rectified linear unit (ReLU) activation functions [29]. As shown in Fig. 2, input to the DQN is a 2-dimensional state modeled using validation loss and WER of wav2vec2.0 training, and output action space is 3-dimensional.

5. Results and Discussion

5.1. Baselines

We observe in Table 1 that for 10 minutes of fine-tuning data, the baseline trained with fixed OAR of 2 outperforms the remaining baselines along with standard wav2vec2.0 trained with OAR of 0 (i.e. without augmentation). This indicates the usefulness of augmentation in wav2vec2.0 training. Similarly, in the case of 1 hour of fine-tuning data, the baseline trained with fixed OAR of 1 outperforms the remaining baselines on 3 out of 4 evaluation sets. This might indicate that a lesser amount of augmented data is required as the amount of training data increases. This phenomenon persists across the 10 hours and 100 hours models as well, where a fixed OAR of 1 yields the optimal result. Moreover, we observe in Table 1 that our baseline system, trained with fixed OAR with LibriSpeech 100 hours, outperforms the corresponding open-source wav2vec2.0 pre-trained ASR on `dev-clean` and `test-clean` evaluation sets with on an average 5.5% of relative improvement.

Further, observations of Table 2 reveal that baselines trained with a fixed OAR of 3 outperform those trained with other OARs. This trend could be attributed to the initialization of wav2vec2.0 with a pre-trained ASR checkpoint, which has already undergone comprehensive training on the original LibriSpeech 100 hours dataset. Hence, more diversity in training data is expected by the model.

[1] Available as of February 2024: https://dl.fbaipublicfiles.comfairseqwav2vecwav2vec_small.pt

[2] Available as of February 2024: https://dl.fbaipublicfiles.com/fairseq/wav2vec/wav2vec_small_100h.pt

Table 1: *Results (in terms of %WER) with self-supervised wav2vec2.0 Base checkpoint trained on different amounts of labeled data scenarios.* $OAR = 0, 1, 2, 3, 4$ *indicates the fixed OAR throughout the wav2vec2.0 training. Results in boldface indicate the best baseline and proposed ROAR based results.*

		LibriSpeech Evaluation Splits			
Labeled Data	OAR	Dev-Clean	Dev-Other	Test-Clean	Test-Other
10 Minutes	0 (no augmentation)	40.7	48.8	41.5	48.9
	1	39.1	47.3	40.0	48.0
	2	**39.0**	**47.3**	**39.8**	**47.5**
	3	40.1	47.8	40.4	48.0
	4	40.4	48.2	40.8	48.7
	ROAR	**37.15**	**45.9**	**38.11**	**46.0**
1 Hour	0 (no augmentation)	19.5	29.5	20.2	20.4
	1	**19.3**	29.2	**19.9**	**29.8**
	2	19.6	**29.1**	20.0	30.0
	3	19.9	30.2	20.5	30.8
	4	20.2	30.8	21.0	31.3
	ROAR	**18.6**	**28.4**	**19.1**	**28.9**
10 Hours	0 (no augmentation)	9.9	19.3	10.1	19.5
	1	**9.5**	**18.2**	**9.5**	**18.4**
	2	9.6	18.5	9.7	18.5
	3	9.9	18.9	10.0	18.7
	4	10.2	19.4	10.5	19.9
	ROAR	**9.1**	**17.7**	**9.3**	**17.7**
100 Hours	open-source	6.1	13.8	6.1	13.5
	0 (no augmentation)	6.0	14.1	6.1	13.9
	1	5.8	**13.7**	**5.9**	**13.5**
	2	**5.6**	13.8	6.0	13.9
	3	6.2	14.1	6.1	14.0
	4	6.1	14.0	6.0	13.9
	ROAR	**5.3**	**13.3**	**5.6**	**13.1**

Table 2: *Results (in terms of %WER) with pre-trained wav2vec2.0 Base model further trained on LibriSpeech 100 hours training split along with different augmentation strategies. The confidence interval is computed over the models obtained from 5 different runs.*

		LibriSpeech Evaluation Splits			
Labeled Data	OAR	Dev-Clean	Dev-Other	Test-Clean	Test-Other
100 Hours	open-source	6.1	13.8	6.1	13.5
	0	6.1 ± 0.1	13.7 ± 0.1	6.1 ± 0.1	13.4 ± 0.1
	1	6.1 ± 0.1	13.4 ± 0.2	6.1 ± 0.1	13.2 ± 0.2
	2	6.1 ± 0.1	13.5 ± 0.2	6.0 ± 0.1	13.0 ± 0.1
	3	6.0 ± 0.1	13.4 ± 0.3	6.0 ± 0.1	12.9 ± 0.2
	4	6.2 ± 0.2	13.6 ± 0.3	6.1 ± 0.2	13.2 ± 0.3
	ROAR	5.8 ± 0.2	13.1 ± 0.2	5.9 ± 0.1	12.6 ± 0.2

5.2. ROAR Based Wav2Vec2.0

We observe in Table 1 that ROAR based model archives on an average improvement of 3.75%, 3.26%, 3.35%, 4.07% over best baselines (trained with fixed OAR) on 10 minutes, 1 hour, 10 hours, and 100 hours of LibriSpeech, respectively. Further, we observe in Table 2 that the ROAR based model, further trained on wav2vec2.0 pre-trained ASR, outperforms the open-source wav2vec2.0 pre-trained ASR with an average 4.96% as well as the best baseline trained with fixed OAR with an average relative improvement of 2.4% of relative improvement across standard LibriSpeech evaluation sets. This indicates the significance of the dynamic adjustment of OAR in wav2vec2.0-based ASR training.

Moreover, for stability analysis, we also present the results along with confidence interval, over 5 different runs, in Table 2.

5.3. OAR Dynamics

Deep Q-network based OAR dynamics and the evolution of OAR dynamics across further episodes in the context of wav2vec2.0 CTC training for LibriSpeech 1H are illustrated in Fig. 3. We observe in Fig. 3a that OAR fluctuates but remains very small throughout the wav2vec2.0 training. This behavior might be attributed to the random initialization of DQN weights at the beginning of the *episode 1* and the limited interaction of DQN with the wav2vec2.0 training environment. As the training further proceeds, we observe in the OAR dynamics of *episode 2* that the DQN is trying to keep the OAR small at the beginning and end of the wav2vec2.0 training and relatively higher OAR in the mid-iterations of training. This phenomenon is further reinforced in *episode 3*, where OAR is once again small at the outset of training, increases in a stepwise fashion during mid-range iterations, remains elevated throughout this phase and eventually decreases to 0 in a stepwise manner at the end of training. These observations hypothesize our intuition depicted in 1 and claimed in Section 1.

6. Conclusion

This study demonstrates that dynamically adjusting the OAR using DQN provides advantages over the commonly employed fixed OAR approach for wav2vec2.0 based ASR. While the benefit of DQN-based OAR dynamics is evident, our study also introduces a few limitations to the proposed approach. First, the validation WER is expected to decrease at the beginning of ASR training, and hence DQN might not get reinforced very well at the beginning of training. Second, the DQN agents are initialized randomly at the beginning of episode 1, and hence a detailed stability analysis on OAR dynamics will be beneficial.

While in this study, we focus entirely on optimizing the OAR dynamic through DQN, it can be extended by jointly optimizing other augmentation hyperparameters such as modification factors (e.g. signal-to-noise ratio, speed modification factor) along with OAR.

7. Acknowledgment

This work was partially supported by Academy of Finland (Decision No. 349605, project “SPEECHFAKES”)

8. References

[1] J. M. Ramirez, A. Montalvo, and J. R. Calvo, "A survey of the effects of data augmentation for automatic speech recognition systems," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, 2019.

[2] 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," in *Proc. Interspeech*, 2019.

[3] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019.

[4] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Proc. Interspeech*, 2015.

[5] M. Bartelds, N. San, B. McDonnell, D. Jurafsky, and M. Wieling, "Making more of little data: Improving low-resource automatic speech recognition using data augmentation," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 2023.

[6] L. Meng, J. Xu, X. Tan, J. Wang, T. Qin, and B. Xu, "Mixspeech: Data augmentation for low-resource automatic speech recognition," in *Proc. ICASSP*, 2021.

[7] C. Liu, Q. Zhang, X. Zhang, K. Singh, Y. Saraf, and G. Zweig, "Multilingual graphemic hybrid ASR with massive data augmentation," in *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, 2020.

[8] V. P. Singh, H. Sailor, S. Bhattacharya, and A. Pandey, "Spectral modification based data augmentation for improving end-to-end ASR for children's speech," in *Proc. Interspeech*, 2022.

[9] V. P. Singh, M. Sahidullah, and T. Kinnunen, *ChildAugment: Data augmentation methods for zero-resource children's speaker verification*. The Journal of Acoustical Society of America, 2024.

[10] T. K. Lam, S. Schamoni, and S. Riezler, "Make more of your data: Minimal effort data augmentation for automatic speech recognition and translation," in *Proc. ICASSP*. IEEE, 2023, pp. 1–5.

[11] S. Sivasankaran, E. Vincent, and I. Illina, "Discriminative importance weighting of augmented training data for acoustic model training," in *Proc. ICASSP*, 2017.

[12] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[13] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 2018.

[14] D. Haziza, J. Rapin, and G. Synnaeve, "Population based training for data augmentation and regularization in speech recognition," *CoRR*, vol. abs/2010.03899, 2020.

[15] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "AutoAugment: Learning augmentation strategies from data," in *Proc. IEEE CVPR*, 2019.

[16] S. Lim, I. Kim, T. Kim, C. Kim, and S. Kim, "Fast autoaugment," in *Proc. Advances in Neural Information Processing Systems*, 2019.

[17] K. Tian, C. Lin, M. Sun, L. Zhou, J. Yan, and W. Ouyang, "Improving auto-augment via augmentation-wise weight sharing," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 19 088–19 098.

[18] S. Ren, J. Zhang, L. Li, X. Sun, and J. Zhou, "Text AutoAugment: Learning compositional augmentation policy for text classification," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021.

[19] B. Zoph and Q. Le, "Neural architecture search with reinforcement learning," in *Proc. ICLR*, 2017.

[20] 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*, 2020.

[21] C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D. dissertation, King's College, Oxford, 1989.

[22] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, 2015.

[23] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "LibriSpeech: An ASR corpus based on public domain audio books," in *Proc. ICASSP*, 2015.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Advances in Neural Information Processing Systems*, 2017.

[25] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Pro. ICML*, 2020.

[26] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. ICML*, 2006.

[27] D. Snyder, G. Chen, and D. Povey, "MUSAN: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[28] 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. ICASSP*, 2017.

[29] A. F. Agarap, "Deep learning using rectified linear units (ReLU)," *arXiv preprint arXiv:1803.08375*, 2018.