INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

# MixRep: Hidden Representation Mixup for Low-Resource Speech Recognition

*Jiamin Xie, John H.L. Hansen*

Center for Robust Speech Systems (CRSS), University of Texas at Dallas, TX, 75080
{Jiamin.Xie, John.Hansen}@utdallas.edu

## Abstract

In this paper, we present MixRep, a simple and effective data augmentation strategy based on mixup for low-resource ASR. MixRep interpolates the feature dimensions of hidden representations in the neural network that can be applied to both the acoustic feature input and the output of each layer, which generalizes the previous MixSpeech method. Further, we propose to combine the mixup with a regularization along the time axis of the input, which is shown as complementary. We apply MixRep to a Conformer encoder of an E2E LAS architecture trained with a joint CTC loss. We experiment on the WSJ dataset and subsets of the SWB dataset, covering reading and telephony conversational speech. Experimental results show that MixRep consistently outperforms other regularization methods for low-resource ASR. Compared to a strong SpecAugment baseline, MixRep achieves a +6.5% and a +6.7% relative WER reduction on the eval92 set and the Callhome part of the eval'2000 set.
**Index Terms**: End-to-end Speech Recognition, Low-resource, Mixup, Hidden Representations, Data Augmentation

## 1. Introduction

Deep learning research has fueled many recent advancements toward solving the automatic speech recognition (ASR) task. The end-to-end (E2E) ASR [1, 2, 3] predicts the textual output from the time-frequency input by a deep stack of convolutional neural networks (CNN) [4], recurrent neural networks (RNN) [5], or attention layers [6]. The large modeling capacity of the E2E ASR model helps learn a direct mapping from the input to the output sequence effectively, as shown in many works [7, 8]. While large models are powerful to achieve impressive performance [9] given a sizeable training set, they tend to memorize examples and become overly confident with incorrect predictions [10, 11]. For low-resource scenarios, overfitting becomes an issue [12, 13] with other challenges like diverse acoustic variations [14, 15] and language mismatch [16, 17].

Data augmentation is one effective way to expand the training data and make models generalize [18, 19]. Developed techniques for ASR create multiple views of the original speech [20] by applying vocal tract length normalization [21], reverberation [22], and tempo variations [20]. Advanced methods synthesize speech directly using the state-of-the-art text-to-speech [23] and voice conversion [24] models, which is shown beneficial for low-resource distant talks [25]. Other methods like SpecAugment [26] randomly crops and modifies the input spectrogram like images along both time and frequency dimensions. Feature mixup [10, 11] is another angle to create artificial examples by exploring the input space through interpolation, where a mixup refers to the convex combination of two training features. One recent work of ASR studies the mixup between mel-spectrograms of two utterances and trains the E2E model to predict both reference texts from the mixed feature [27].

Since the hidden representation space of an ASR model can encode information (e.g. phoneme, word, and semantics) more abstract than the acoustic features at the input [28, 29], we reason performing the mixup of hidden representations is beneficial. As shown in the previous study [11], the mixup performed at deep layers of a model has regularization effects on the representations. It reduces variations in the dimensions that encode redundant information and also smooths the classification boundaries among representations, which alleviates over-confident predictions for adversarial or ambiguous input. For E2E speech recognition, we hypothesize such regularization would improve the overall learning as the speech input contains many variations caused by low-dimensional factors such as content, speakers, and channels [30].

In this study, we propose a data augmentation method for low-resource ASR based on representation mixup, named MixRep. The contribution of this work is as follows,

1. A data augmentation strategy using the mixup of hidden representations for low-resource speech recognition [1]
2. Highlight of the complementary regularization on both time and frequency (feature) dimensions for mixup methods
3. Investigation of other techniques, e.g. SpecAugment [26] and MixSpeech [27], and their comparison to MixRep

## 2. Related Work

The concept of input mixup [10] has been successfully applied to classification tasks because the labels are one-hot and easy for interpolation, e.g. pictures [31], acoustic scenes [32], speakers [33], etc. For ASR acoustic model training [34], the mixup is conducted for the HMM state labels aligned to the speech input. For tasks with label sequences of different lengths, the mixup of training losses is used instead, e.g. for the E2E model training in speech recognition [27] or machine translation [35]. The Manifold Mixup [11] extends input mixup to the hidden representations of a deep neural network, which is the focus of our study. For speech input, this has only been previously studied for sound classification [36] and one recent work on speech translation [37], where the latter applies mixup to representations from two modalities and does not consider a mixup of target sequences. Unlike previous work, we investigate the application of Manifold Mixup to train an E2E ASR model. We intend to learn the behavior of different layers, so we do not search layer combinations as extensively as done in [36]. Our approach is similar to the MixSpeech [27] method but extends it and explores the combination of techniques.

[1] https://github.com/jiamin1013/mixrep-espnet


10.21437/Interspeech.2023-1216

# 3. Method

In this section, we first review the mixup [11] concept. We then explain the MixSpeech method [27] that applies mixup to E2E ASR. Finally, we describe our proposed method which extends the speech mixup to the hidden representation, and mention its regularization effect on the feature dimension.

### 3.1. Manifold Mixup

The Manifold Mixup [11] is a generalized version of the input mixup [10] that allows representation ouput from any layer of a neural network model to be linearly interpolated (i.e. mixup). For an arbitrary $K$-layers model, we denote $f_{n,k}(\cdot)$ the underlying function that processes data from the $n$-th layer input to the $k$-th layer output, where $n = 0$ is the model input and $f_{0,0}(\cdot)$ is the identity function. Suppose a supervised learning task has input features $X$ and one-hot labels $Y$, the Manifold Mixup trains the model by mixing up the hidden representations and labels,

$$R_k = \lambda * f_{0,k}(X_i) + (1 - \lambda) * f_{0,k}(X_j), \quad (1)$$

$$Y_{mix} = \lambda * Y_i + (1 - \lambda) * Y_j, \quad (2)$$

$$\mathcal{L}_{mix} = \mathcal{L}(f_{k,K}(R_k), Y_{mix}), \quad (3)$$

where $\lambda \in [0, 1] \sim Beta(\alpha, \alpha)$ with $\alpha \in (0, \infty)$ and $i$ and $j$ denote two training examples. The interpolation results in a new training example represented by the hidden dimensions of the model, thus it is an effective data augmentation method. We note the input mixup [10] becomes a special case of the Manifold Mixup [11] when $n$ and $k$ are both 0.

### 3.2. MixSpeech: Input Mixup

MixSpeech [27] is a data augmentation method developed for E2E ASR training based on the input mixup [10]. For a pair of utterances, this method mixes up acoustic features of these utterances in the frequency dimensions frame-by-frame. Because speech input and text output have different lengths with the alignment unknown, mixing two word labels at the same position does not correspond to a simultaneous time when both words are spoken. So, the MixSpeech interpolates the losses of recognizing each textual label sequence instead.

### 3.3. MixRep: Hidden Representation Mixup

We propose MixRep to create artificial examples during training by mixing hidden representations of an E2E ASR model, inspired by the previous methods [11, 27]. Reusing $R_k$ defined in Equation 1, MixRep interpolates sampled utterances $i$ and $j$ frame-by-frame by their respective output from the $k$-th layer of a model. For the textual label sequences $Y$, MixRep trains the model to optimize the following loss,

$$\begin{aligned}\mathcal{L}_{mixRep} = \lambda * \mathcal{L}(f_{k,K}(R_k), Y_i) \\ + (1 - \lambda) * \mathcal{L}(f_{k,K}(R_k), Y_j),\end{aligned} \quad (4)$$

where $k$ is drawn uniformly from a set of eligible layers $S$ on each forward pass. When $k = 0$, since the hidden representations are mel-spectrograms from the input, MixRep naturally extends the MixSpeech [27] method. We present the detailed steps of our proposed method in Algorithm 1.

One key aspect of the mixup methods [10, 11] is their regularization benefits on the feature dimension, aside from data augmentation. By making the interpolation weight in the mixup of features and that of the reference labels match, the method constructs a linear association between the input and output space of the neural network [10]. For Manifold Mixup [11], the linearity is constructed for the hidden representation space. This has shown to regularize the feature dimensions of the hidden representations by capturing salient low-dimensional variations and enforcing smooth classification boundaries for predictions made on the representations. Because MixRep regularizes the representation space but speech contains both time and frequency information, we propose the following two configurations of the MixRep method:

- *Basic*: does not apply any regularization along the time axis of the input, similar to [27]
- *Time enhanced*: applies regularization along the time axis of the input (e.g. time masking or warping, etc.).

To explore the *Time enhanced* approach, we investigate applying regularization to the input (line 18 of Algorithm 1). For deep layers of the model (a large *k*), the representation encodes much information due to a large receptive field. Masking representations at a deep layer then impacts performance since the masked content can be hardly recovered by the limited modeling capacity which follows. In order to recognize the missing content from masking, applying time regularization to the input is effective for helping the following attention-based layers to learn strong representation that captures meaning than fine details from the input. We consider it is crucial for MixRep since a good hidden representation space needs to be established.

**Algorithm 1** Hidden Representation Mixup (MixRep)

1: **Given** a subset $\mathcal{S} \in \{0, 1, \ldots, K\}$, a beta coefficient $\alpha$, a pre-processing function $m(\cdot)$
2: **procedure** MIXUP($x, y, \lambda$)
3: get $batchSize$ from $x$
4: $indArr \leftarrow$ shuffle list $[0, 1, \ldots, batchSize - 1]$
5: $x \leftarrow \lambda * x + (1 - \lambda) * x[indArr, :]$ *// interpolation*
6: $\tilde{y} \leftarrow y[indArr, :]$
7: **return** $x, \tilde{y}$
8: **end procedure**
9: **for** *each batch* **do**
10: $\lambda \sim Beta(\alpha, \alpha)$ *// sample an interpolation weight*
11: $k \sim Uniform(\mathcal{S})$ *// sample a layer index*
12: $x \leftarrow batch$
13: **for** $(index,\ layer)$ in $layers$ **do**
14: **if** $index = k$ **then**
15: $x,\ \tilde{y} \leftarrow$ MIXUP($x, y, \lambda$)
16: **end if**
17: **if** $index = 0$ **then**
18: $x \leftarrow m(x)$ *// for masked-based preprocessing*
19: **end if**
20: $x \leftarrow layer.forward(x)$
21: **end for**
22: backward $loss \leftarrow \lambda * \mathcal{L}(x, y) + (1 - \lambda) * \mathcal{L}(x, \tilde{y})$
23: **end for**

# 4. Experimental Setup

To examine the effectiveness of MixRep, we conduct experiments on ASR benchmarks that evaluate speech from reading newspapers or conversations over the telephone. For the Conformer architecture illustrated in section 4.2, we mix representations from the output of an encoder layer (i.e. after the final LayerNorm [38]) and use the original positional encoding without mixup. We establish SpecAugment [26] as our baselines, which randomly and partially masks out time and frequency content

from the input. By mixing the input acoustic feature, we recreate the MixSpeech [27] method. For fair comparisons, we test both the *Basic* and *Time enhanced* configurations of these methods in our experiments. We then apply the best configuration to mix representations and compare the performance of MixRep to the SpecAugment baseline and the effective MixSpeech.

### 4.1. Datasets

The Wall Street Journal (WSJ) [39] and Switchboard (SWB) [40] datasets are investigated in our study. The WSJ dataset includes read speech with transcripts drawn from the newspaper. The data is partitioned into 81 hours of training speech (*si284*), 1 hour for development (*dev93*), and 0.7 hour for evaluation (*eval92*). The SWB dataset contains spontaneous speech from two sides of a conversation over the telephone line. To simulate a low-resource setup, we randomly sample the training data into two subsets totaling 40 hours and 80 hours. We use the single-fold train split without any speed or noise perturbation. We use the eval'2000 (LDC2002S09) dataset as evaluation for SWB, where there are Switchboard (swb) and Callhome (chm) parts that are unseen from the SWB training/validation set.

### 4.2. E2E ASR model

For ASR experiments, we follow recipes provided in the ESPnet toolkit [41] to train an E2E ASR model for each dataset, which is further referred to as the *Default* setup. Our models use the listen, attend, and spell (LAS) architecture [1] that include the Conformer encoder [38] and the Transformer [42] decoder. We extract 80 mel-filterbanks and 3-dimensional pitch features. The input is then passed through an optional SpecAugment [26], followed by 2D-CNNs with a downsampling factor of 4. The SpecAugment uses time warping with a window size of 5, two frequency masks with $F = 30$, and two time masks with $T = 40$, unless otherwise stated. The encoder has 12 layers. The decoder has 6 layers and connects to a softmax layer followed by the cross-entropy (CE) loss. The model is trained jointly by $L_{joint} = \alpha * L_{ctc} + (1 - \alpha) * L_{ce}$ [43], where $\alpha$ is set to $0.3$ in our study. The label smoothing weight is $0.1$. The model dimension is 256. The attention modules have 4 attention heads and 2048 linear units with a dropout $p = 0.1$. We use the warmup learning rate scheduler for all datasets. The learning rate of WSJ peaks at $0.005$ after 30k steps and that of SWB peaks at $0.006$ after 25k steps. We use character as output to train the WSJ model and byte-pair-encoding (bpe) with 2000 subword units [2] for the SWB model. The number of elements in a batch is 2.5M for WSJ and 10M for SWB. The gradients accumulation is 6 times. We use a CNN kernel size of 15 for WSJ and 31 for SWB. The WSJ is trained for 150 epochs and 300 epochs for SWB. Both experiments finish in 1 day using two or four 2080Ti GPUs.

### 4.3. Parameters of MixRep

We use the beta distribution with a coefficient $\alpha = 2$ for all experiments using MixRep. This corresponds to a convex-shaped probability distribution with mean equals 0.5 (i.e. $E[\lambda] = 0.5$) and about half of the probability mass (56%) falls between 0.3 and 0.7. Following MixSpeech [27], we also use $\tau = 0.15$ for WSJ (means 15% data of a batch uses the mixup), but we find $\tau = 0.45$ to be more suitable for SWB. Since searching all subsets of the layers in the ASR encoder is infeasible (i.e. $2^{12} = 4096$ combinations), we employ the following heuristic: we first apply MixRep to every single layer of the ASR encoder and gather its performance; we then test the set $S$ containing the best-performing layer and the input layer. We report every single-layer performance in section 5.4.

[2] The bpe model is obtained from texts in full SWB training

## 5. Results

### 5.1. Baselines and Previous Methods

Because the ESPnet default setting includes the SpecAugment, we expect it to be the best and make it the baseline. To make a fair comparison to the *Time enhanced* configuration, we investigate turning off frequency masking for SpecAugment. The original MixSpeech is applied to the Transformer model, so we recreate their method for the Conformer model. The results of these systems are illustrated in Table 1.

Table 1: *WER of baselines and previous methods. T and F refer to SpecAugment regularization along the time and frequency fimension, respectively. Default setup is explained in Section 4.*

| Dataset | Model | T | F | With LM (%) dev | With LM (%) eval | No LM (%) dev | No LM (%) eval |
|---|---|---|---|---|---|---|---|
| WSJ | **Transformer** | | | | | | |
| | Espnet [41] | ✓ | ✓ | 7.4 | 4.9 | - | - |
| | MixSpeech [27] | ✗ | ✗ | - | 4.7 | - | - |
| | **Conformer** | | | | | | |
| | Default | ✓ | ✓ | 7.1 | 4.7 | 11.2 | 8.9 |
| | Default | ✓ | ✗ | **6.2** | 4.3 | 10.4 | 7.7 |
| | + MixSpeech (Ours) | ✗ | ✗ | 6.8 | 4.5 | 10.7 | 8.4 |
| | + MixSpeech (Ours) | ✓ | ✗ | 6.3 | **4.2** | **9.8** | **7.5** |
| SWB 40hr | **Conformer** | | | | | | |
| | Default | ✓ | ✓ | - | - | 21.3 | 34.1 |
| | Default | ✓ | ✗ | - | - | **18.5** | 31.6 |
| | + MixSpeech (Ours) | ✗ | ✗ | - | - | 20.7 | 33.0 |
| | + MixSpeech (Ours) | ✓ | ✗ | - | - | 18.9 | **30.6** |
| SWB 80hr | **Conformer** | | | | | | |
| | Default | ✓ | ✓ | - | - | 13.5 | 23.3 |
| | Default | ✓ | ✗ | - | - | 13.2 | 23.3 |
| | + MixSpeech (Ours) | ✗ | ✗ | - | - | 14.7 | 25.2 |
| | + MixSpeech (Ours) | ✓ | ✗ | - | - | **13.0** | **22.6** |

From Table 1, we can observe the frequency content from the input is critical for low-resource setups. Comparing the SpecAugment configurations within the default setups, turning off frequency masking improves performance overall. This shows less significantly in the SWB 80hr setup (the model still improves on the in-domain set, but stagnates on the out-of-domain one). Comparing our MixSpeech setups, we observe the benefit of regularization on the time axis for the mixup. There is at least 7% relative improvement on the evaluation sets across all datasets, which verifies our hypothesis on the benefits of regularization on the time axis for mixup-based methods (see Section 3.3). Finally, we turn off frequency masking in baselines and use *Time enhanced* configuration for MixRep.

### 5.2. Read English speech

We compare MixRep to the best baseline and the input mixup for read English ASR. The results of MixRep applied at each layer are displayed in Figure 1. The experimental results are illustrated in Table 2.

From Figure 1, we observe mixing up in the deep layers (layer 7 to 10) gives good improvements over the baseline. This finding somewhat corresponds to the previous study [28], which finds middle to deep layers of a CNN-RNN E2E ASR model

Table 2: *WER on the WSJ corpus of proposed MixRep method. S denotes the set of layers to be selected from (see Section 3.3).*

| Model | With LM (%) | | No LM (%) | |
|---|---|---|---|---|
| | dev93 | eval92 | dev93 | eval92 |
| **Conformer** | | | | |
| SpecAug. baseline | 6.2 | 4.3 | 10.4 | 7.7 |
| + MixRep $S = \{0\}$ | 6.3 | 4.2 | 9.8 | 7.5 |
| + MixRep $S = \{9\}$ | 6.1 | **4.1** | **9.4** | **7.2** |
| + MixRep $S = \{0, 9\}$ | **6.0** | 4.2 | 9.8 | 7.5 |

trained on LibriSpeech contain more phonetic information than the early to middle layers. We hypothesize that certain layers of the E2E ASR model encode information similar to the output textual space, thus applying MixRep helps enforce this association by the linear relationship imposed.

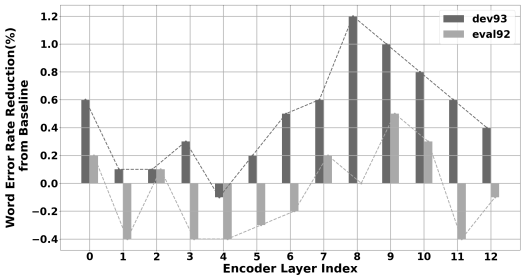


Figure 1: *Per-layer improvement of MixRep compared to the SpecAugment baseline on the WSJ corpus.*

We observe a superior performance using MixRep from the results presented in Table 3. Mixing up the 9-th layer representations outperforms the SpecAugment baseline by $+6.5\%$ relative and the input mixup by $+4\%$ on the evaluation set. When decoding with the LM, the improvement is diminished slightly, suggesting the benefits of the mixup may come from learning more linguistic knowledge in the encoder representations.

### 5.3. Spontaneous telephony speech

We compare MixRep to other regularization methods for spontaneous telephony ASR. The results of MixRep applied at each layer are displayed in Figure 2. The experimental results are illustrated in Table 3.

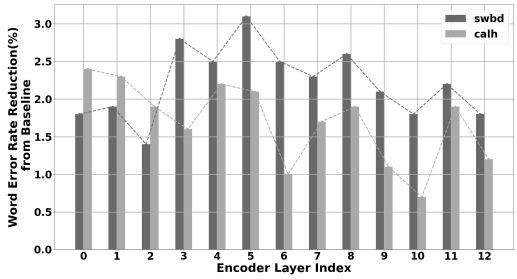


Figure 2: *Per-layer improvement of MixRep compared to the SpecAugment baseline on the eval2000 using 40 hours of SWB.*

From Figure 2, we observe MixRep achieves significant and consistent gains over the SpecAugment baseline on the 40 hours SWB, which proves MixRep to be an effective method for low-resource training. Moreover, layer 5, being the strongest performance on average, improves over the input mixup at the 0-th layer. Compared to Figure 1, we notice stronger improvements obtained by mixing up early to middle layer for the spontaneous telephony speech. Moreover, we spot a similar downward trend from layer 8 to layer 12, suggesting $\{8\}$ or $\{9\}$ can be a safe choice for the hyperparameter $S$.

Table 3: *WER on the eval'2000 using 40- and 80 hours training data subsets from the SWB corpus of proposed MixRep method.*

| Train Data | Model | With LM (%) | | No LM (%) | |
|---|---|---|---|---|---|
| | | swb | chm | swb | chm |
| SWB 40hr | **Conformer** | | | | |
| | SpecAug. baseline | 16.8 | 29.6 | 18.5 | 31.6 |
| | + MixRep $S = \{0\}$ | 17.1 | 28.4 | 18.9 | 30.6 |
| | + MixRep $S = \{5\}$ | **16.1** | 29.1 | **17.6** | 30.9 |
| | + MixRep $S = \{0, 5\}$ | 16.3 | **27.7** | 17.7 | **29.5** |
| SWB 80hr | SpecAug. baseline | 12.0 | 21.8 | 13.2 | 23.3 |
| | + MixRep $S = \{0\}$ | 12.1 | **21.1** | 13.0 | 22.6 |
| | + MixRep $S = \{0, 5\}$ | 11.9 | 21.3 | **12.8** | 22.8 |
| | + MixRep $S = \{0, 9\}$ | **11.8** | 21.2 | **12.8** | **22.5** |

For the SWB 40hr dataset in Table 3, we verify applying MixRep to multiple layers can achieve better performance than a single layer. Mixing up both the 0-th layer and 5-th layer representations outperforms the SpecAugment baseline by a $+6.6\%$ relative on the Callhome set, suggesting complementary learning behavior upon regularizing multiple layers for ASR. This is similar to the previous finding for sound classification [36]. For the SWB 80hr dataset in Table 3, we observe the impact of training data size. The MixRep $S = \{0, 5\}$ configuration leads the baseline by a +2.1% relative after the training data is doubled. This verifies the data augmentation aspect of MixRep, but also shows the limitation of performance gain when the training data becomes sufficient. On the other hand, using the set $S = \{0, 9\}$ outperforms $S = \{0, 5\}$, which indicates the heuristic to select the optimal set $S$ is not optimal and is open for future work.

## 6. Conclusions

In conclusion, we presented MixRep in this paper, a method to create artificial examples by interpolating hidden representations for E2E ASR training. We proposed an enhanced strategy for mixup-based methods, where a regularization along the time axis at the input is added. This is shown to be complementary to the feature regularization effect of the mixup for ASR. By experimenting on both read and spontaneous telephony styles of speech, we showed a significant and consistent improvement of MixRep over other regularization techniques such as SpecAugment and MixSpeech for low-resource ASR. We discussed the impact of training data size and the heuristic for searching the optimal set of eligible layers, which opens up future work.

## 7. Acknowledgements

The authors would like to thank Szu-Jui Chen for the meaningful discussion and suggestions on the work.

## 8. References

[1] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *IEEE ICASSP'16*, pp. 4960–4964.

[2] A. Graves and A. Graves, "Connectionist temporal classification," *Supervised sequence labelling with recurrent neural networks*, 2012.

[3] A. Graves, "Sequence transduction with recurrent neural networks," *ICML 2012 Workshop on Representation Learning*.

[4] T. N. Sainath, B. Kingsbury, A.-r. Mohamed, G. E. Dahl, G. Saon, H. Soltau, T. Beran, A. Y. Aravkin, and B. Ramabhadran, "Improvements to deep convolutional neural networks for lvcsr," in *IEEE ASRU'13*, pp. 315–320.

[5] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *IEEE ICASSP'13*.

[6] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *IEEE ICASSP'16*, pp. 4945–4949.

[7] W. Chan, C. Saharia, G. Hinton, M. Norouzi, and N. Jaitly, "Imputer: Sequence modelling via imputation and dynamic programming," in *ICML'20*, pp. 1403–1413.

[8] S. Toshniwal, T. N. Sainath, R. J. Weiss, B. Li, P. Moreno, E. Weinstein, and K. Rao, "Multilingual speech recognition with a single end-to-end model," in *IEEE ICASSP'18*, pp. 4904–4908.

[9] Z. Tüske, G. Saon, and B. Kingsbury, "On the Limit of English Conversational Speech Recognition," in *Proc. Interspeech 2021*, 2021, pp. 2062–2066.

[10] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *ICLR'18*.

[11] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, D. Lopez-Paz, and Y. Bengio, "Manifold mixup: Better representations by interpolating hidden states," in *International conference on machine learning*. PMLR, 2019, pp. 6438–6447.

[12] Y. Sharma, B. Abraham, K. Taneja, and P. Jyothi, "Improving Low Resource Code-Switched ASR Using Augmented Code-Switched TTS," in *Proc. Interspeech 2020*, 2020, pp. 4771–4775.

[13] K. Peterson, A. Tong, and Y. Yu, "OpenASR20: An Open Challenge for Automatic Speech Recognition of Conversational Telephone Speech in Low-Resource Languages," in *Interspeech'21*.

[14] J. H. Hansen, A. Joglekar, S.-J. Chen, M. Chandra Shekar, and C. Belitz, "Fearless steps APOLLO: Advanced naturalistic corpora development," in *LREC 2022*.

[15] S.-J. Chen, W. Xia, and J. H. L. Hansen, "Scenario aware speech recognition: Advancements for apollo fearless steps & chime-4 corpora," *IEEE ASRU'21*, pp. 289–295.

[16] A. Rouhe, A. Virkkunen, J. Leinonen, and M. Kurimo, "Low Resource Comparison of Attention-based and Hybrid ASR Exploiting wav2vec 2.0," in *Proc. Interspeech 2022*, pp. 3543–3547.

[17] E. Morris, R. Jimerson, and E. Prud'hommeaux, "One Size Does Not Fit All in Resource-Constrained ASR," in *Interspeech'21*.

[18] S. Wu, H. Zhang, G. Valiant, and C. Ré, "On the generalization effects of linear transformations in data augmentation," in *ICML'20*.

[19] W.-N. Hsu, A. Sriram, A. Baevski, T. Likhomanenko, Q. Xu, V. Pratap, J. Kahn, A. Lee, R. Collobert, G. Synnaeve, and M. Auli, "Robust wav2vec 2.0: Analyzing Domain Shift in Self-Supervised Pre-Training," in *Interspeech-21*, pp. 721–725.

[20] 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.

[21] 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.

[22] 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 *IEEE ICASSP'17*, pp. 5220–5224.

[23] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural speech synthesis with transformer network," in *AAAI'19*.

[24] T. Kaneko and H. Kameoka, "Cyclegan-vc: Non-parallel voice conversion using cycle-consistent adversarial networks," in *IEEE EUSIPCO'18*, pp. 2100–2104.

[25] E. Tsunoo, K. Shibata, C. Narisetty, Y. Kashiwagi, and S. Watanabe, "Data Augmentation Methods for End-to-End Speech Recognition on Distant-Talk Scenarios," in *Proc. Interspeech 2021*.

[26] 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*, 2019, pp. 2613–2617.

[27] L. Meng, J. Xu, X. Tan, J. Wang, T. Qin, and B. Xu, "Mixspeech: Data augmentation for low-resource automatic speech recognition," in *IEEE ICASSP'21*.

[28] Y. Belinkov and J. Glass, "Analyzing hidden representations in end-to-end automatic speech recognition systems," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[29] C.-Y. Li, P.-C. Yuan, and H.-Y. Lee, "What does a network layer hear? analyzing hidden representations of end-to-end asr through speech synthesis," in *IEEE ICASSP'20*, pp. 6434–6438.

[30] D. Wang, L. Deng, Y. T. Yeung, X. Chen, X. Liu, and H. Meng, "VQMIVC: Vector Quantization and Mutual Information-Based Unsupervised Speech Representation Disentanglement for One-Shot Voice Conversion," in *Proc. Interspeech 2021*, 2021, pp. 1344–1348.

[31] D. Wang, Y. Li, L. Wang, and B. Gong, "Neural networks are more productive teachers than human raters: Active mixup for data-efficient knowledge distillation from a blackbox model," in *CVPR'20*.

[32] L. D. Pham, I. McLoughlin, H. Phan, and R. Palaniappan, "A robust framework for acoustic scene classification." in *INTERSPEECH*, 2019, pp. 3634–3638.

[33] Y. Zhu, T. Ko, and B. Mak, "Mixup learning strategies for text-independent speaker verification." in *Interspeech*, 2019, pp. 4345–4349.

[34] I. Medennikov, Y. Y. Khokhlov, A. Romanenko, D. Popov, N. A. Tomashenko, I. Sorokin, and A. Zatvornitskiy, "An investigation of mixup training strategies for acoustic models in asr." in *Interspeech*, 2018, pp. 2903–2907.

[35] D. Guo, Y. Kim, and A. Rush, "Sequence-level mixed sample data augmentation," in *EMNLP'20*.

[36] A. Jindal, N. E. Ranganatha, A. Didolkar, A. G. Chowdhury, D. Jin, R. Sawhney, and R. R. Shah, "SpeechMix — Augmenting Deep Sound Recognition Using Hidden Space Interpolations," in *Interspeech'20*.

[37] Q. Fang, R. Ye, L. Li, Y. Feng, and M. Wang, "STEMM: Self-learning with speech-text manifold mixup for speech translation," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 7050–7062. [Online]. Available: https://aclanthology.org/2022.acl-long.486

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

[39] D. B. Paul and J. Baker, "The design for the wall street journal-based csr corpus," in *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*, 1992.

[40] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "Switchboard: Telephone speech corpus for research and development," in *IEEE ICASSP'92*.

[41] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-End Speech Processing Toolkit," in *Proc. Interspeech 2018*, 2018.

[42] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[43] S. Kim, T. Hori, and S. Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *IEEE ICASSP'17*.