CONTINUAL LEARNING USING LATTICE-FREE MMI FOR SPEECH RECOGNITION

Hossein Hadian, Arseniy Gorin

Behavox, Montreal, Canada

ABSTRACT

Continual learning (CL), or domain expansion, recently became a popular topic for automatic speech recognition (ASR) acoustic modeling because practical systems have to be updated frequently in order to work robustly on types of speech not observed during initial training. While sequential adaptation allows tuning a system to a new domain, it may result in performance degradation on the old domains due to catastrophic forgetting. In this work we explore regularization-based CL for neural network acoustic models trained with the lattice-free maximum mutual information (LF-MMI) criterion. We simulate domain expansion by incrementally adapting the acoustic model on different public datasets that include several accents and speaking styles. We investigate two well-known CL techniques, *elastic weight consolidation* (EWC) and *learning without forgetting* (LWF), which aim to reduce forgetting by preserving model weights or network outputs. We additionally introduce a sequence-level LWF regularization, which exploits posteriors from the denominator graph of LF-MMI to further reduce forgetting. Empirical results show that the proposed sequence-level LWF can improve the best average word error rate across all domains by up to 9.4% relative compared with using regular LWF.

Index Terms— Automatic Speech Recognition, Continual Learning, Domain Expansion, Lattice-free MMI, Learning Without Forgetting

1. INTRODUCTION

Commercial automatic speech recognition (ASR) systems must frequently work robustly with a variety of accents, recording devices, background noises, and speech styles. While training the acoustic models on thousands [1, 2] and hundreds of thousands [3] of hours of diverse speech helps the model generalize better, it still needs to be updated from time to time to cover domains that have not been sufficiently represented in the training data. Similar to [4] we define domain as a group of utterances that share some common characteristics. Retraining the model with new data from scratch is expensive and difficult if privacy limitations do not allow keeping target domain data and the initial training data in one place. In practice, adaptation of the acoustic models using target domain speech appears to be a more favorable approach [5]. One problem is that the adapted model's performance on previously learned data may significantly degrade due to catastrophic forgetting [6, 7], which limits the model's constant improvement across multiple domains.

Continual learning (CL) or life-long learning [8], which was intensively studied especially in computer vision, is a variety of machine learning methods that aim to learn new tasks without forgetting previously acquired knowledge. Broadly, CL methods are based on regularization techniques [9, 10, 11], dynamic neural network architectures [12, 13, 14], and rehearsal (also called memory replay) [12, 15, 16].

Overcoming catastrophic forgetting when adapting the ASR acoustic models to new domains can be seen as a special case of CL, as the model usually keeps doing the same task (predicting the same set of classes). Authors of [4] defined this task as domain expansion and explored several CL regularization techniques to optimize the ASR performance on initial and adapted speech accents. Later, [17] explored similar regularization techniques to sequentially train a multi-dialect ASR system. Alternatively, [18] proposed a model expansion–based approach, which demonstrated good results but at a price of growing the model size with the number of training tasks, where tasks are defined as public datasets used by the authors. Similarly, [19] explored a combination of model parameters and network outputs as a method for longitudinal training – a scenario in which the model learns a single domain but with incrementally appearing data and without access to historical data. [20] compared regularization- and rehearsal-based techniques for incrementally learning an end-to-end CTC-based ASR system. They demonstrated better performance of rehearsal technique that, however, requires storing some portion of the training data from each domain.

Similar to [17], we focus on two popular regularization-based techniques, namely, elastic weight consolidation (EWC) [11] and learning without forgetting (LWF) [9]. We do not consider rehearsal-based techniques as in general they require access to the source data during adaptation and therefore are hard or impossible to use in privacy-preserving settings. Dynamic architectures are also not considered as these methods generally result in growing the model size over time, eventually slowing down the inference.

In this work[1] we apply EWC and LWF on acoustic models trained using the LF-MMI [21] criterion and assess their efficacy. Furthermore, as the main contribution of this work, we propose a sequence-level version of LWF regularization that is based on LF-MMI and we demonstrate its effectiveness through multiple experiments. The experiments are conducted on several public databases that involve several English accents (US, Indian, Singapore), speech styles (read speech, public talks, and conversations) and channels (studio and telephone).

2. LATTICE-FREE MMI

Lattice-free MMI is a method of training an acoustic model using the MMI objective function without generating utterance-specific denominator lattices [21]. More specifically, there is only one denominator graph that is used for all utterances during training. The objective function is as follows[2]:

$$\mathcal{F}_{MMI} = \log \frac{p_\lambda(\boldsymbol{x}|\mathbb{M}_{\boldsymbol{w}})}{p_\lambda(\boldsymbol{x})} \tag{1}$$

---

[1]https://github.com/kaldi-asr/kaldi/blob/master/egs/cl_english/v1/run.sh.

[2]Note that in this and the rest of the equations in this paper, for brevity we show the objective functions for one utterance only. The full objective function would be the sum over all utterances. Also note that an objective function is **maximized** during training.

The denominator can be approximated as:

$$p_\lambda(\boldsymbol{x}) = \sum_l p_\lambda(\boldsymbol{x}|\mathbb{M}_l) \approx p_\lambda(\boldsymbol{x}|\mathbb{M}_{den}) \qquad (2)$$

where $\mathbb{M}_{den}$ is a finite-state transducer (FST) graph that includes all possible sequences of words. This is called the denominator graph, as opposed to $\mathbb{M}_{\boldsymbol{w}}$, which is called the numerator graph and is utterance specific. Expanding the log gives us:

$$\mathcal{F}_{MMI} = \log p_\lambda(\boldsymbol{x}|\mathbb{M}_{\boldsymbol{w}}) - \log p_\lambda(\boldsymbol{x}|\mathbb{M}_{den}) \qquad (3)$$

Note that the first term is maximizing the likelihood of the paths in the numerator graph which represents the true label and the second term is minimizing the likelihood of all other paths in the denominator graph.

3. CONTINUAL LEARNING

The two most popular regularization-based approaches for CL (LWF and EWC) address the forgetting problem by forcing the network to preserve either its outputs or its parameters, respectively. The following subsections briefly describe these approaches in more detail.

3.1. Learning Without Forgetting

The LWF objective function is a cross-entropy term that is added as a regularization term to the main objective function for training the neural network:

$$\mathcal{F}_{LWF} = \alpha \sum_i y_i^{src} \log(y_i) \qquad (4)$$

where y_i are the outputs of the network while training and y_i^{src} are the pre-computed outputs of the network using the source model (i.e., the previous step of CL) for all of the target domain training examples. We may simply call the y_i^{src} reference posteriors in this context. The scale α controls the effect of the LWF regularization.

3.2. Elastic Weight Consolidation

While LWF concerns remembering the output distribution, EWC is focused on the parameters of the network. Specifically, the EWC objective function is as follows:

$$\mathcal{F}_{EWC} = -\alpha \sum_d \sum_j F_j^d (\theta_j^d - \theta_j)^2 \qquad (5)$$

where d loops over previous domains, and F_j^d is the Fisher diagonal value for the j_{th} parameter of the network estimated using the data and model at CL step d. Also θ_j is the j_{th} parameter of the current network and θ_j^d is the corresponding parameter from the model at CL step d. Note that similar to [17] we use the online computation approach for the Fisher matrix estimation. Also note that while LWF relies only on the output of CL at the previous domain, EWC needs statistics (i.e., Fisher matrices) from all previous steps.

4. PROPOSED SEQUENCE LEVEL OBJECTIVE

As mentioned earlier, the LWF loss is cross-entropy, which is frame-level. However, LF-MMI is a sequence-level loss function [21]. It appears sub-optimal to train the acoustic model with a sequence-level objective and apply LWF using a frame-level objective. Therefore, instead of using the neural network outputs as reference posteriors for LWF, we propose to use the denominator posteriors (i.e.,

occupancies) as the reference posteriors. This leads to the following term for the LWF objective function:

$$\mathcal{F} = \alpha \sum_i \gamma_i^{src} \log(y_i) \qquad (6)$$

where γ_i^{src} are the denominator posteriors from the previous domain. These posteriors are computed by running the forward-backward algorithm on the denominator graph given the neural network outputs and are therefore sequence-level. However, this cross-entropy term poses a problem because the target distribution (i.e., γ^{src}) is sequence-level whereas the predicted distribution (i.e., the neural network outputs while training: y_i) is frame-level. This results in excessive boosting of the denominator posteriors as we are treating them as targets.[3] To offset this effect, we subtract the denominator log-prob term scaled by the same LWF scale α. This gives us the proposed sequence-level LWF objective function:

$$\mathcal{F}_{DenLWF} = \alpha \sum_i \gamma_i^{src} \log(y_i) - \alpha \log p_\lambda(\boldsymbol{x}|\mathbb{M}_{den}) \qquad (7)$$

Taking the derivative of this term with respect to log outputs leads to the following term:

$$\frac{\partial \mathcal{F}_{DenLWF}}{\partial \log(y_i)} = \alpha(\gamma_i^{src} - \gamma_i) \qquad (8)$$

where the γ_i are the denominator occupancies during training. We can see immediately from this equation that the gradients are 0 at the beginning of training since the training denominator posteriors match the source ones. This is a desired characteristic, as in the beginning there is no forgetting so we do not expect any gradients from the LWF regularization term. The same cannot be said about Equation 6, which leads to an unbalanced derivative.

5. EXPERIMENTAL SETUP

We choose 5 different English datasets (summarized in Table 1) for sequential training.

Name	Dataset	Speech type	English accent	Size, hours
FSH	Fisher English [22] subset	CTS[4]	US	200
LIB	Librispeech [23] subset	Read	Mostly US	50
TED	TED-LIUM [24] subset	Talks	Mostly US	50
NSC	National Speech Corpus [25] (IRV subset)	CTS	Singaporean	50
CVI	Mozilla Common Voice [26] v6.1 (Indian speakers)	Read	Indian	40

Table 1: Summary of datasets used in the experiments.

A 50-hour subset from each domain is used for domain expansion, except for CVI, for which there is only 40 hours of speech available. The 200-hour FSH dataset is used to train the seed model in all experiments. For each dataset, evaluation is done on their official dev/test subsets. For LIB, a combination of dev_clean and test_clean subsets is used and for TED, a combination of dev and test subsets. A 3-gram language model is estimated using the training text of all domains and is used in all experiments. We do not use i-vectors or other kinds of speaker adaptation in training the acoustic models. In the following experiments, FT refers to a model that

---

[3]Note that if we try to train the network with this term, catastrophic forgetting is not prevented (results are not shown).

Model	FSH	TED	LIB	NSC	CVI	Avg
Seed (FSH)	18.2	15.6	11.8	56.3	58.4	32.1
Comb	18.1	11.8	8.3	25.3	25.0	17.7

Table 2: Results for the seed model and the combined-data model.

has been trained using naive fine-tuning and therefore is considered as the worst-case baseline of catastrophic forgetting. On the other hand, $Comb$ refers to a model trained on all the available data from all domains combined [5]. Table 2 summarizes the WERs for the seed and $comb$ models on all domains. It is clear that the most challenging domains for this seed model are CVI and NSC.

Kaldi ASR toolkit is used for all experiments [27]. The network is made up of 14 TDNN-F layers with a total of 14M parameters [28] and is trained using LF-MMI with mini-batch gradient descent. The seed model's decision tree is used during all steps of domain expansion. The alignments at each step are generated using a Gaussian mixture model (GMM) trained on the target domain at that step.

Similar to [17], we also compute gap recovery rates for the CL steps as follows:

$$\text{Gap Recovery} = 1.0 - \frac{\text{WER}_{CL} - \text{WER}_{Comb}}{\text{WER}_{FT} - \text{WER}_{Comb}} \qquad (9)$$

where WER_{CL}, WER_{Comb}, and WER_{FT} refer to WERs of the model trained using the CL method, the $comb$ model, and the FT model, respectively. In addition to gap recovery rates, at each step we also define *Relative Forgetting*, which measures the degradation of WER on the past domains as a result of that step, and *Relative Learning*, which measures the improvement of WER on the target domain at the end of that step. Formally, they are defined as follows:

$$\text{Relative Learning} = 1.0 - \frac{\text{CWER}}{\text{CWER}^{src}} \qquad (10)$$

$$\text{Relative Forgetting} = \frac{\text{PWER}}{\text{PWER}^{src}} - 1.0 \qquad (11)$$

where CWER is the WER on the current domain at each step (e.g., at step TED → LIB, it would be the WER on the LIB test set at the end of the step). CWER^{src} is the WER on the same dataset, but using the model at the end of the previous step. PWER and PWER^{src} are similar to CWER and CWER^{src} except they are measured on past domains (averaged). For example, at step TED → LIB of the sequence FSH → TED → LIB, the past domains are FSH and TED.

6. EXPERIMENTS AND RESULTS

This section describes a series of experiments to assess the efficacy of LWF, EWC, and the proposed method on LF-MMI trained acoustic models. Specifically, FSH is used to create the seed model, after which we do 4 steps of CL in the following order: FSH → TED → LIB → NSC → CVI. The detailed results are listed in Table 3. Note that after each step, only the domains involved in training up to that step are evaluated. We can see that naive FT leads to the best performance on target data but severely degrades performance on all previous domains. After the first step, EWC outperforms LWF. However, in the subsequent steps LWF performs considerably better, especially in the last 2 steps. This is consistent with observations of [17]. The results also show that DenLWF achieves a WER similar to that of LWF in the second step and remarkably better WERs in the other steps. Specifically, for the last two steps, which are the most challenging, DenLWF relatively improves LWF's average

---

WER by 3% and 9.4%, respectively. If we look at the individual domain WERs, the improvements are even more visible. For example, in the last step, while DenLWF achieves a slightly worse (but close) WER on CVI, it achieves a much better WER on previous domains. In particular, DenLWF improves LWF's WER on FSH from 39.7% to 33.0% and on TED from 23.7% to 19.7%.

It should be noted that all methods were tuned across multiple combinations of source-target transfers. For LWF, we use a scale of $\alpha = 1.0$ while for DenLWF a lower scale of $\alpha = 0.6$ works best. For EWC, after normalizing the Fisher diagonal values to have a median of 1.0, we use $\alpha = 300$.
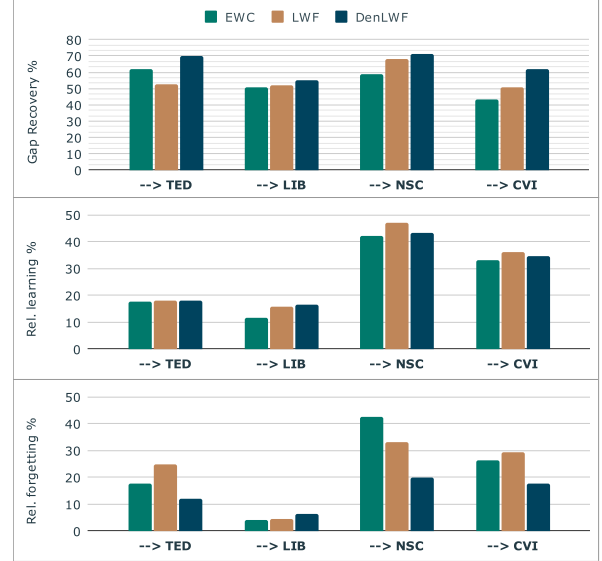
Fig. 1: Overall performance of different methods. The upper chart shows average gap recovery percentages for different steps of the CL training pipeline. The middle and lower charts show relative learning and forgetting percentages, respectively.

Figure 1 shows the gap recovery and relative learning/forgetting percentages (as defined in Equations 9-11) for the steps in Table 3. In the top plot we can see that DenLWF achieves higher gap recovery rates in all steps, especially in the first and last steps. In the second step, all methods have a similar performance, which might be attributable to the fact that LIB is an easy domain for our seed model. Specifically, EWC and LWF achieve a gap recovery rate of 40-60% and 50-68%, respectively, in different steps of CL, while DenLWF achieves a gap recovery rate of 55-71% for these steps. In the middle and lower charts we see the relative learning and forgetting rates, respectively. DenLWF and LWF show a similar capacity to learn new domains, with LWF learning slightly more in the third step. However, as can be seen in the lower chart, this comes at the cost of much more forgetting for LWF. Specifically, LWF has a relative forgetting rate of around 30% in the last two steps whereas DenLWF only forgets around 20% from the past domains.

6.1. Analysis and Discussion

In order to look more closely into how the proposed DenLWF functions and how it compares with regular LWF, we do two experiments. In the first experiment, we measure and plot the regular LWF objective function (i.e., Equation 4) when training a model using regular LWF as well as DenLWF. Note that LWF cross-entropy objective

---

[5]Note that unlike [17], we do not train separate $comb$ models at each step as there is negligible difference in WER.

| Method | FSH | → TED | | | → LIB | | | | → NSC | | | | | → CVI | | | | | |
	FSH	FSH	TED	Avg	FSH	TED	LIB	Avg	FSH	TED	LIB	NSC	Avg	FSH	TED	LIB	NSC	CVI	Avg
FT	18.2	30.6	13.3	21.9	29.0	16.7	9.0	18.2	48.4	42.3	33.9	27.6	38.0	59.5	42.2	23.5	58.0	27.9	42.2
EWC	18.2	21.7	13.5	17.6	22.1	14.4	9.7	15.4	29.9	20.6	15.5	34.0	25.0	38.4	25.3	15.4	47.0	31.7	31.6
LWF	18.2	23.0	13.5	18.2	22.8	14.0	9.2	15.3	30.5	18.3	12.5	30.5	23.0	39.7	23.7	14.0	41.7	30.0	29.8
DenLWF	18.2	20.7	13.4	17.1	22.5	13.9	9.1	15.2	27.5	16.2	10.9	34.5	22.3	33.0	19.7	12.0	40.0	30.3	27.0

Table 3: WERs,% of 4 CL steps using different baseline and proposed methods. The first column shows the seed model. The next 3 columns under "→ TED" are for the first step FSH → TED, the columns under "→ LIB" are for the second step TED → LIB, and so forth.
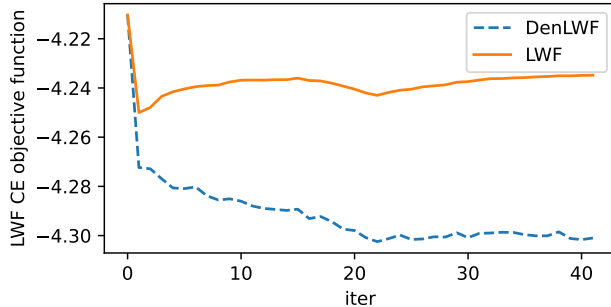
Fig. 2: Comparing LWF cross-entropy objective during training for regular (frame-level) LWF and proposed DenLWF.

for DenLWF is computed here only for visualization purposes and is not used in back-propagation. The resulting plot is shown in Figure 2. It can be observed that for regular LWF, the objective function degrades right after the first iteration of training and then starts to improve as training progresses. This effectively restricts the output distribution from diverging too far from the source domain posteriors. However, in the case of DenLWF, we can see that the output posteriors keep diverging from the reference posteriors until halfway through training, at which point they plateau. This seems to allow the network to learn better, as the network output is under weaker frame-level restriction compared to regular LWF. In other words, regular LWF forces the network to remember the source domain on a frame-level basis, whereas the proposed DenLWF objective forces the network to remember the source domain on a sequence-level basis – which is more consistent with the speech signal – leading to better overall performance (i.e., average WER).

In the second experiment, we investigate the impact of the LWF scale α on the average WER as well as on individual WERs. Figure 3 compares regular LWF and DenLWF when trained using different scales' α (all other hyperparameters are the same). The training step used in this experiment is FSH → CVI. The upper plot shows the average WER (on FSH and CVI) while the lower plots show the WERs on FSH and CVI separately. It can be seen that regular LWF achieves its best performance when the scale is close to 1.0; whereas DenLWF achieves its best performance when we use a smaller scale – closer to 0.6. This suggests that DenLWF is a more powerful regularizing term that needs a smaller scale. We can also see that, generally, DenLWF achieves much less forgetting (i.e., lower WER on FSH), which comes at the cost of less learning (i.e., higher WER on CVI). However, at the scales where DenLWF works best (i.e., 0.5 and 0.6), the difference in CVI WER between DenLWF and LWF is quite small. Note that this plot also shows that for all the scales tested, the worst-case performance of DenLWF is better than the best-case
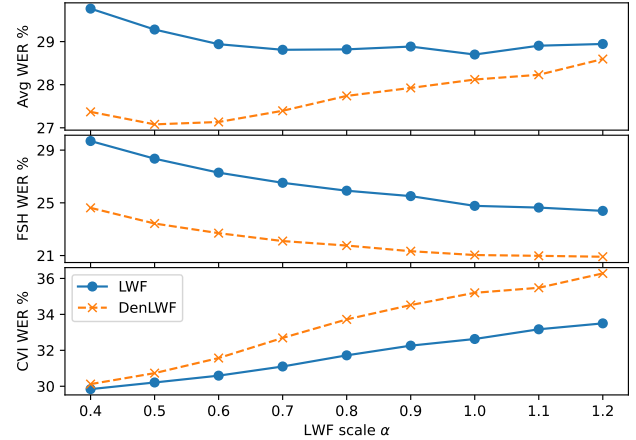
Fig. 3: Average and individual WERs% for regular LWF and DenLWF on FSH → CVI transfer.

performance of LWF on the FSH → CVI step. We observed similar plots for other steps (not shown due to lack of space).

7. CONCLUSIONS

In this study, we applied two well-known regularization-based methods for continual learning, namely, EWC and LWF, to the state-of-the-art lattice-free MMI acoustic models and investigated their efficacy through a series of experiments. We observed gap recovery rates of 40-60% using EWC and 50-68% using LWF, for different steps of CL. Compared to naive fine-tuning, EWC improved the final average WER (after all the steps of CL) from 42.2% to 31.6%, while LWF improved it to 29.8%. In addition, given the fact that lattice-free MMI is a sequence-level objective function, we proposed a new LWF objective function called DenLWF where we use MMI's denominator posteriors as reference posteriors in LWF while adding an offsetting term. Through experiments we showed that DenLWF improved the gap recovery rates to 55-71% for different steps of CL. Also, DenLWF improved the final average WER from 29.8% down to 27.0%.

8. REFERENCES

[1] Tatiana Likhomanenko et al., "Rethinking evaluation in ASR: Are our models robust enough?," in *INTERSPEECH*. ISCA, 2021.

[2] William Chan et al., "Speechstew: Simply mix all available speech recognition data to train one large neural network," *arXiv preprint arXiv:2104.02133*, 2021.

[3] Arun Narayanan et al., "Toward domain-invariant speech recognition via large scale training," in *SLT*. IEEE, 2018.

[4] Shahram Ghorbani, Soheil Khorram, and John HL Hansen, "Domain expansion in DNN-based acoustic models for robust speech recognition," in *ASRU*. IEEE, 2019.

[5] Peter Bell et al., "Adaptation algorithms for neural network-based speech recognition: An overview," *IEEE Open Journal of Signal Processing*, vol. 2, pp. 33–66, 2020.

[6] Michael McCloskey and Neal J Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of learning and motivation*, vol. 24, pp. 109–165. Elsevier, 1989.

[7] Robert M French, "Catastrophic forgetting in connectionist networks," *Trends in cognitive sciences*, vol. 3, no. 4, pp. 128–135, 1999.

[8] German I Parisi et al., "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, 2019.

[9] Zhizhong Li and Derek Hoiem, "Learning without forgetting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[10] Friedemann Zenke, Ben Poole, and Surya Ganguli, "Continual learning through synaptic intelligence," in *International Conference on Machine Learning*. PMLR, 2017, pp. 3987–3995.

[11] James Kirkpatrick et al., "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[12] Tyler L Hayes, Nathan D Cahill, and Christopher Kanan, "Memory efficient experience replay for streaming learning," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 9769–9776.

[13] Andrei A Rusu et al., "Progressive neural networks," *arXiv preprint arXiv:1606.04671*, 2016.

[14] Jaehong Yoon et al., "Lifelong learning with dynamically expandable networks," in *ICLR*, 2018.

[15] David Lopez-Paz and Marc'Aurelio Ranzato, "Gradient episodic memory for continual learning," *Advances in neural information processing systems*, vol. 30, pp. 6467–6476, 2017.

[16] Geoffrey E Hinton and David C Plaut, "Using fast weights to deblur old memories," in *Proceedings of the 9th Annual Conference of the Cognitive Science Society*, 1987, pp. 177–186.

[17] Brady Houston and Katrin Kirchhoff, "Continual learning for multi-dialect acoustic models.," in *INTERSPEECH*. ISCA, 2020.

[18] Samik Sadhu and Hynek Hermansky, "Continual learning in automatic speech recognition.," in *INTERSPEECH*. ISCA, 2020.

[19] Joachim Fainberg, *Robust learning of acoustic representations from diverse speech data*, Ph.D. thesis, The University of Edinburgh, 2021.

[20] Heng-Jui Chang, Hung-yi Lee, and Lin-shan Lee, "Towards Lifelong Learning of End-to-end ASR," in *INTERSPEECH*. ISCA, 2021.

[21] Daniel Povey et al., "Purely sequence-trained neural networks for ASR based on lattice-free MMI.," in *INTERSPEECH*. ISCA, 2016.

[22] Christopher Cieri, David Miller, and Kevin Walker, "The Fisher corpus: A resource for the next generations of speech-to-text.," in *LREC*, 2004.

[23] Vassil Panayotov et al., "Librispeech: an ASR corpus based on public domain audio books," in *ICASSP*. IEEE, 2015.

[24] Anthony Rousseau, Paul Deléglise, Yannick Esteve, et al., "Enhancing the TED-LIUM corpus with selected data for language modeling and more TED talks," in *LREC*, 2014.

[25] Jia Xin Koh et al., "Building the Singapore English national speech corpus," in *INTERSPEECH*, 2019.

[26] R. Ardila et al., "Common voice: A massively-multilingual speech corpus," in *LREC*, 2020.

[27] Daniel Povey et al., "The kaldi speech recognition toolkit," in *ASRU*. IEEE, 2011.

[28] Daniel Povey et al., "Semi-orthogonal low-rank matrix factorization for deep neural networks.," in *INTERSPEECH*. ISCA, 2018.