Unsupervised Online Continual Learning for Automatic Speech Recognition

Steven Vander Eeckt, Hugo Van hamme

KU Leuven
Department Electrical Engineering ESAT-PSI, Leuven, Belgium
{steven.vandereeckt, hugo.vanhamme}@esat.kuleuven.be

Abstract

Adapting Automatic Speech Recognition (ASR) models to new domains leads to Catastrophic Forgetting (CF) of previously learned information. This paper addresses CF in the challenging context of Online Continual Learning (OCL), with tasks presented as a continuous data stream with unknown boundaries. We extend OCL for ASR into the unsupervised realm, by leveraging self-training (ST) to facilitate unsupervised adaptation, enabling models to adapt continually without label dependency and without forgetting previous knowledge. Through comparative analysis of various OCL and ST methods across two domain adaptation experiments, we show that UOCL suffers from significantly less forgetting compared to supervised OCL, allowing UOCL methods to approach the performance levels of supervised OCL. Our proposed UOCL extensions further boosts UOCL's efficacy. Our findings represent a significant step towards continually adaptable ASR systems, capable of leveraging unlabeled data across diverse domains.

Index Terms: automatic speech recognition, online continual learning, unsupervised continual learning, self-training

1. Introduction

Artificial Neural Networks (ANN) have propelled Automatic Speech Recognition (ASR) to new heights. Despite their advancements, ANNs are hindered by Catastrophic Forgetting (CF) [1], which severely limits the capacity of ASR models to learn continually. Fortunately, Continual Learning (CL) has emerged as a solution to overcome CF in ANNs, showing considerable progress recently. CL enables models to learn from all accessible data, thus building powerful models that perform robustly across various accents, dialects, speakers, and domains in ASR, without requiring the reintroduction of all past data to prevent forgetting of old knowledge. The progress in CL includes, since a few years, also ASR [2, 3, 4, 5, 6, 7, 8, 9, 10].

The more challenging scenario within CL is Online Continual Learning (OCL). Unlike offline CL, OCL involves: (i) unknown task boundaries; (ii) a continuous stream of batches from unidentified tasks; and (iii) a single pass over each batch, i.e. accesss to a batch is lost after learning from it. In contrast to offline CL's sequential task presentation, OCL assumes a non-i.i.d. stream of batches, whose task is unknown to the model, that must be utilized without impairing previous task performance. OCL for ASR remains relatively unexplored, with only a handful of studies addressing it [7, 8, 10].

This paper extends the realm of OCL into unsupervised OCL (UOCL) for ASR. UOCL challenges the model to learn from unlabeled batches from new (and unknown) tasks while preserving performance on previously encountered domains. We adopt self-training, where the model generates its own pseudo-labels, to facilitate learning from unlabeled data. Our contributions are as follows: (1) we delve into the problem of UOCL, with a focus on the interplay between forgetting and self-training; (2) we compare various OCL and self-training methods; and (3) we propose a new UOCL method, building upon the work in [10]. Our findings reveal that UOCL experiences significantly less forgetting compared to supervised OCL, often resulting in UOCL methods closely matching the performance of supervised OCL approaches due to the balance between reduced forgetting in UOCL and the effectiveness of learning from ground truth vs. pseudo-labels in supervised OCL. Our proposed extension enhances UOCL performance further, demonstrating a significant reduction in the performance gap with its supervised counterparts, nearly equaling performance of supervised scenarios. This advancement marks a considerable step towards General CL [11], aiming to enable ASR models to leverage all available data to continually improve without forgetting, resulting in highly robust and versatile models.

2. Framework & Prior Research

2.1. Problem Statement

Online CL begins with a model defined by parameters θ_0, trained on a labeled dataset $\mathcal{D}_0$ for task (e.g. certain accent, dialect, text domain, etc.) $\mathcal{T}_0$. Upon deployment, the model processes a continuous, non-i.i.d. stream of data batches $\{\mathcal{B}_i\}_{i>0}$, each related to a specific task $\mathcal{T}_{t_i}$, but without knowledge of which task a batch belongs to. As the model processes a new batch $\mathcal{B}_{i+1}$ with parameters θ_i, it seeks to learn as much as possible, updating to θ_{i+1}, while ensuring it does not forget its knowledge from $\mathcal{T}_0$ or previous batches $\{\mathcal{B}_j\}_{j\in(0,i]}$. Access to the initial data, $\mathcal{D}_0$, and earlier batches is no longer available.

In contrast to earlier OCL research in ASR, which primarily considers batches of labeled data pairs (X, y) [8, 10], this paper, similar to [7], explores unsupervised OCL, moving beyond the need for labeled data and focusing on scenarios with only unlabeled speech X. Our objective is for the model to effectively use unlabeled data, enhancing its performance on all tasks encountered $\mathcal{T}_k$ for $k \geq 0$, while not degrading (and ideally even improving) on past tasks. To learn from unlabeled data, self-training is considered.

2.2. Related Work

While UOCL has been explored within computer vision [12, 13, 14], within the context of ASR, it remains, with the exception of [7], a largely unexplored area. UOCL builds upon various domains investigated in ASR.

Online Continual Learning. Though offline CL has garnered

significant interest in ASR [2, 4, 5, 6, 9], exploration of the more demanding online CL scenario remains limited [7, 8, 10]. Our contribution builds upon the existing work by further advancing OCL into the realm of unsupervised learning.

Self-Training (ST). ST [15] leverages a model initially trained on labeled data to generate pseudo-labels (PLs) for unlabeled data, facilitating subsequent model updates. These PLs can be produced through various methods such as greedy CTC [16, 17], beam search decoding [18, 19], or by incorporating an external LM [20, 21]. ST thus allows the same or a new model to be updated on the pseudo-labeled data.

Unsupervised Domain Adaptation (UDA). UDA involves adapting a model from a source domain to a target domain using unlabeled data. In the context of ASR, examples include [22, 23, 24]. Although UDA shares similarities with UOCL, it diverges in several key aspects: it assumes that (a) labeled data from the source domain remains available; (b) the model has access to the full unlabeled target training set; (c) only the performance on the target domain matters.

Test-Time Adaptation (TTA). TTA occurs when the source data is unavailable while adapting to new domains using (unlabeled) test data, often limited in quantity, such as a batch or a single utterance [25]. Works for ASR include [25, 26]. Despite similarities to UOCL, TTA differs significantly: (a) UOCL involves adapting to multiple tasks or domains without forgetting, unlike TTA, which focuses on immediate adaptation with minimal data; (b) TTA adaptations are performed at test time, often based on limited data, whereas OCL adapts during training, handling more extensive tasks.

3. Methodology

UOCL combines Self-Training (ST) for generating pseudo-labels with OCL strategies to counteract forgetting. After outlining the model, alongside the ST and OCL methods pertinent to our investigation, we present tailored enhancements to an established OCL technique, optimizing its efficacy for UOCL.

3.1. Model

The model considered in this study is an encoder-decoder end-to-end ASR model, characterized by parameters $\theta \in \mathbb{R}^N$. It takes speech X as input, consisting of F speech frames. The model outputs a probability distribution across C word pieces. Incorporating a hybrid approach [27], our model integrates losses from both CTC and a decoder, weighted by c and $1-c$, resp., with $c \in [0, 1]$. The model contains (during training) random effects such as dropout [28] and data augmentation.

3.2. Self-Training Methods

We explore these ST methods to generate pseudo-labels (PLs):
(Greedy) CTC. The decoder is not involved in PL generation, which is fast and only based on acoustic evidence, not on a (strong) internal LM in the decoder.
Hybrid. Combines decoder and CTC to produce PLs, with same weight c as during training and beam size b as during testing.
BERT-CTC (BCTC). BERT-CTC [29] integrates BERT's [30] linguistic knowledge into non-autoregressive CTC predictions for ASR. This approach mitigates the conditional independence assumption by leveraging external LM knowledge, enhancing adaptability across varying text domains without biasing towards the initial task's domain. This ST method should be particularly suited for experiments with text domain shifts. We train the BERT-CTC on top of θ_0 (which we freeze) on task $\mathcal{T}_0$. During UOCL, to update model θ_i on batch $\mathcal{B}_{i+1}$, we add BERT-CTC (now frozen) on top of θ_i to generate PLs.

An external LM trained on new tasks' text domains is not explored, assuming these text domains are unknown a priori. When generating PLs, the model's random effects are disabled.

3.3. Online CL Methods

We consider the following OCL methods to apply to UOCL:
Fine-Tuning (FT). The model is naively adapted to the new tasks without mechanism to alleviate forgetting. Consequently, Fine-Tuning is expected to suffer greatly from forgetting.
Experience Replay (ER) [31]. ER alleviates forgetting by relying on a small memory set of samples of old tasks, from which it samples a mini-batch to train jointly with the current batch; i.e. samples from old tasks are used to regularize updates to the model. We consider the implementation of [3] with reservoir sampling [32] to fill the memory with a size of M utterances. For unlabeled data, the speech and PLs are stored in the memory set. Despite its effectiveness, ER's requirement for data storage can be a major drawback when data storage is restricted.
Averaging for Online CL of ASR (AOS) [10]. AOS employs a dual-model approach: a continually updated adapted model and a stable final model. The adapted model learns from new data batches, with its updates incrementally averaged into the final model. Specifically, after updating the adapted model $\tilde{\theta}_i$ with batch $\mathcal{B}_{i+1}$ to $\tilde{\theta}_{i+1}$, the final model θ_i updates to $\theta_{i+1} = (1-\eta_{i+1})\theta_i + \eta_{i+1}\tilde{\theta}_{i+1}$. The weight η_{i+1} for the encoder is based on number of frames in current batch F_{i+1}, total seen frames $F_{1:i}$, and hyper-parameter τ as $\eta_{\text{enc},i+1} = \tau F_i/(F_{1:i} + \tau F_i)$; similar for the decoder, based instead on number of output tokens and using hyper-parameter τ_2. (τ, τ_2) are used to control model plasticity. Finally, a knowledge distillation (KD) from final to adapted model is used to regularize the latter. For UOCL adaptation, we thus let the final model generate PLs for the adapted model's training, replacing the KD regularization previously used to temper the adapted model's updates.

3.4. Our Proposed UOCL Extensions

Building on AOS's proven effectiveness in bypassing the need for past data storage [10], we adapt it for UOCL, creating AOS-U (AOS-Unsupervised), with simple yet effective modifications:

(1) The adapted model is our default choice for generating PLs, as it learns quickly, enhancing the quality of PLs rapidly.
(2) Despite utilizing the adapted model for PL generation, we still omit the KD regularization to avoid impeding its learning progress, ensuring optimal PL performance.
(3) Before updating, the model processes each batch multiple times ($K > 1$), utilizing randomness (dropout, data augmentation) to enhance adaptation from unlabeled data. This approach, while still updating the model only once per batch, enriches learning by presenting varied data representations. We take $K = 2$.

4. Experiments

All experiments were done in ESPnet2 [33]. For all detailed information, we refer to our Github repository [1].

Data. We consider two experiments: (*Exp. 1*) Adaptation from Librispeech-360h (LIB) [34] to six Libri-Adapt (LIB-APT) [35]

[1] https://github.com/StevenVdEeckt/unsupervised-ocl-for-asr

Table 1: *Results of the experiments. Tasks are learned from left to right, with WERs obtained after learning all tasks.* ***S****,* ***U*** *and* ✗*in first column refer to, resp., supervised OCL, UOCL and no adaptation. Best UOCL result per task in bold.*

OCL		Method	ST	Exp. 1: LIB → LIB-APT								Exp. 2: LIB-TED-CV			
				WER per task							Avg.	WER per task			Avg.
				LIB	GB/M	US/U	IN/U	IN/M	US/M	GB/U	WER	LIB	TED	CV	WER
✗	0	Initial model θ_0		6.8	18.4	8.2	20.0	31.7	10.9	9.5	13.18	6.4	14.0	18.8	13.08
S	1	FT		8.1	11.5	8.7	15.6	23.0	11.1	6.4	10.55[g]	8.4	12.5	16.0	12.33[g]
	2	ER [M=2k]		7.1	10.4	7.7	**12.7**	17.9	9.8	6.7	9.05[g]	7.6	12.1	16.0	11.88[g]
	3	AOS		7.2	11.7	7.5	13.1	18.9	9.5	7.4	9.41[g]	6.7	12.5	16.1	11.75[g]
U	4	FT	CTC	7.9	12.1	8.5	16.4	23.5	10.8	6.9	10.76[a]	7.6	12.9	17.1	12.54[a]
	5		Hybr.	8.1	11.7	8.6	16.1	23.2	11.1	**6.9**	10.72[a]	7.6	13.4	17.1	12.70[a]
	6		BCTC	*Excluded due to lack of text-domain shift*								7.7	**12.7**	17.0	12.44[b]
	7	ER [M=2k]	CTC	**7.1**	11.6	7.7	14.1	19.6	9.8	7.3	9.64	6.9	12.9	17.0	12.28
	8		Hybr.	7.2	**11.2**	7.8	14.1	**19.0**	9.9	7.3	9.56[b]	6.8	12.8	17.0	12.21[c]
	9	AOS	CTC	7.2	13.1	**7.5**	14.8	21.5	9.5	7.8	10.17[a]	6.6	13.2	17.2	12.33
	10		Hybr.	7.2	12.9	**7.5**	14.9	21.6	9.6	7.8	10.17[a]	6.8	13.7	19.5	13.32
	11	AOS-U	CTC	7.2	11.9	7.6	**13.6**	19.1	**9.3**	7.5	**9.52**[a,d]	6.6	12.9	17.0	12.15[a,e]
	12		Hybr.	7.2	12.0	**7.5**	13.7	19.3	**9.3**	7.5	9.55[a,d]	6.6	13.5	17.0	12.36[a]
	13		BCTC	*Excluded due to lack of text-domain shift*								**6.5**	12.8	**16.8**	**12.01**[b,f]

[a] Variations in performance across given ST methods for given UOCL method are not statistically significant.
[b, c] Significantly outperforms other ST methods for given UOCL method with level *** for *b* and * for *c*.
[d, e] Significantly outperforms all *other* UOCL methods (with any ST) with level *** (*d*) / at least level ** (*e*), except ER + Hybr (ns).
[f] Significantly outperforms all combinations of UOCL and ST methods with level ***.
[g] Significantly outperforms all UOCL variations of given (supervised) OCL method with level ***.

tasks (9k-30k utterances each), showcasing variations in microphone (USB [U], Matrix [M]) and accent (US [US], Indian [IN], British [GB] English); (*Exp. 2*) LIB to TEDLIUM-3 (TED) [36] to Common Voice-English (CV) [37], learned in this order, considering a subset from TED (100k utterances) and three accents from CV (135k utterances) – this path underscores a text domain shift, marked by CV's distinct shorter sentences with large unique vocabulary, and accent shifts, plus TED's more spontaneous speech.

Model. We consider two models. For Exp. 1, starting from MFCC features, we use 12 Conformer [38] encoder and 6 Transformer [39] decoder layers, each with dimension 2048 and 4 attention heads with dimension 256. For Exp. 2, we start from pre-trained HuBERT [40] features, using HuBERT-Large trained on Libri-Light [41] from [40] as frontend of a model with 6 Conformer encoder and 6 Transformer decoder layers. A linear pre-encoder is added to map the HuBERT features to the same dimension (80) as the MFCC features. The HuBERT model is frozen, while the linear pre-encoder is only trained on the initial task $\mathcal{T}_0$. For both models, the weight of CTC is $c = 0.3$, we use SentencePiece [42] to generate $C = 5000$ output tokens on initial task $\mathcal{T}_0$ and SpecAugment [43] is used as data augmentation while dropout rate is set to $p = 0.1$. Both models (with, resp., $N = 47$M and $N = 349$M [of which 32M trainable] parameters) are trained for 80 epochs on $\mathcal{T}_0$. For OCL, batch size is 10 and 20 for Exp. 1 and Exp. 2, resp., and a SGD optimizer with learning rate 0.01 is used. To evaluate the model, we use the same weight $c = 0.3$ for CTC in addition to beam search decoding with beam size $b = 1$. For BERT-CTC, we set number of iterations to 5. We consider BERT-CTC only for Exp. 2, in which there is a text domain shift.

Hyper-parameters. We optimized each method's hyper-parameters by identifying their best values on a preliminary 'test' experiment, involving LIB-APT and CV tasks (for Exp. 1 and Exp. 2, resp.) not featured in our main experiments.

Metrics. We report WER (in %, evaluated after processing all batches) per task and averaged over all tasks. Using Wilcoxon signed-rank test on the number of errors per utterance [44], we perform significance testing on average WER, considering significance levels $\alpha = 0.05$ (*), $\alpha = 0.01$ (**) and $\alpha = 0.001$ (***), or ns for non-significant ($\alpha > 0.05$).

5. Results

Table 1 presents the results of the two experiments.

5.1. Exp. 1: LIB to LIB-APT

Supervised methods. Compared to the *Initial model*, all supervised methods enhance task performance but exhibit varying degrees of forgetting. ER, benefiting from the storage of past data, achieves the best performance, while AOS, though worse than ER, significantly improves upon the naive FT.

Improvement on all tasks with CTC PL. Without labels, the UOCL methods manage to enhance performance across all tasks relative to the *Initial model*, including on *IN/M* with a high initial WER. This improvement persists despite the model encountering each batch only once and generating its own PLs.

Less forgetting for UOCL with FT. In the UOCL setting, FT using CTC PL exhibits reduced forgetting compared to supervised FT, surpassing it on LIB, US/U, and US/M (for which supervised FT forgets more) and almost matching its average WER. This suggests that for FT, less forgetting from self-training in UOCL nearly counterbalance the superior learning from ground truth labels in supervised OCL. This is not the case for ER and AOS, which mitigate forgetting and for which the

absence of ground truths is thus a larger disadvantage; hence the larger gap with their supervised counterpart.

CTC vs. Hybrid PL. Using Hybrid PL, which includes an internal LM tailored to the new tasks' text domain (the same as $\mathcal{T}_0$), does not significantly improve performance compared to the simpler and faster CTC PL. As its results on LIB, US/U, and US/M illustrate, Hybrid PL tends to forget more than CTC PL. This likely happens because the encoder plays a crucial role in the UOCL process, both in learning new tasks and in reducing forgetting. The decoder, positioned later in the process, can adjust some CTC predictions with its internal LM, which might help to learn the new tasks but also might cause more forgetting (when these PLs are used in the CTC loss). These effects seem to cancel each other out. To explore this idea, we experimented with FT, allowing CTC and decoder to generate and learn from their own PLs, i.e., the PLs made by the decoder are only used in the decoder loss. This results in average WER of 10.61, a significant improvement over CTC and Hybrid PL for FT, combining the advantages from both. Its WERs on LIB, US/U and US/M, are, resp., 7.9, 8.4 and 10.6, illustrating less forgetting. Note, finally, that ER is the only UOCL method whose Hybrid PL significantly outperforms CTC PL. Thanks to its access to past (labeled) data, ER is able to profit from the advantages of Hybrid PL while mitigating the disadvantages.

AOS-U outperforms AOS. Our enhancements to AOS for the UOCL scenario (AOS-U) significantly boost unsupervised AOS's effectiveness, narrowing the performance difference with its supervised counterpart. AOS-U shows improvements in five of the seven tasks, excelling in tasks with initially high error rates (IN/U and IN/M). Although there was a notable performance gap between ER and AOS in both supervised and unsupervised settings, AOS-U matches ER's unsupervised performance without the need to store past data. An ablation study in Table 2 reveals AOS-U's strengths come from both applying the adapted model for pseudo-labeling and processing each batch through the model multiple times.

5.2. Exp. 2: LIB to CV

Less forgetting for UOCL. Unlike in the previous experiment, when adapting LIB to TED and CV, there exist a text domain shift, which causes forgetting in supervised OCL for all methods, though to different extents. When instead learning from PLs (using CTC PLs) in UOCL, we observe a significant reduction in forgetting, not only for the naive FT but also for ER and AOS(-U). For FT, the disadvantage of learning from noisy PLs and the advantage of resulting reduced forgetting almost balance each other out, comparing its performance for UOCL vs. supervised OCL. As in Exp. 1, for ER and AOS(-U), since they mitigate forgetting, the aforementioned disadvantage outweighs the advantage, and, consequently, the gap between their UOCL and supervised OCL performance is larger.

CTC vs. Hybrid ST Method. Even more than in the previous experiment, we observe that, though Hybrid PLs are on average better than CTC PLs, Hybrid ST method suffers from problems which cause its performance to be worse, not better, than CTC. In this experiment, we observe that Hybrid PL, in exceptional cases (1-5% of utterances), tends to generate PLs with very little acoustic evidence, due to e.g. repetitions. This is the case when adapting to both TED (for which utterances are sometimes cut-off sentences) and CV (for which utterances are much shorter). CTC PL thus seems to be more robust for UOCL, even if on average its PLs have higher WER than Hybrid PL. Its advantage is that its predictions are based on direct acoustic evidence, which also results in less forgetting in UOCL.

Table 2: *Ablation study assessing impact of enhancements detailed in Sec. 3.4 for Experiments 1 and 2.*

Method	Average WER	
	Exp. 1	**Exp. 2**
AOS w. CTC-PL	10.17	12.33
+ (1) + (2)	9.70***	12.17***
+ (1) + (2) + (3)	9.52***	12.15^{ns}
+ (1) + (3)	9.49^{ns}	12.20^{ns}

AOS-U outperforms AOS for UOCL. As in the previous experiment, AOS-U significantly outperforms AOS, though in this case, for CTC PL, it is only the usage of the adapted model as generator of PLs that is significant, as illustrated by Table 2.

BERT-CTC PL outperforms other ST Methods. As explained in Sec. 3.2, BERT-CTC, starting from the CTC prediction, allows the model to rely on external LM information without having to make assumptions about the text domain. While the decoder, which relies on an internal LM, has a bias towards the old text domain – no longer valid for the new tasks – BERT-CTC suffers much less from biases. While BERT-CTC decoding using *Initial model* θ_0 results in higher WER on the initial task $\mathcal{T}_0$ compared to Hybrid decoding (4.9 vs. 5.4 on LIB's *dev-other*), its WER on the unseen tasks is better. Consequently, it generates better PLs which result in better overall performance. Compared to Hybrid PL, BERT-CTC PL starts directly from CTC and encoder features, which can attend to BERT features, generating improvements over CTC PL for which there exists more acoustic evidence compared to Hybrid PL. Increased forgetting due to using external information from BERT is minimal, as shown by FT (AOS-U even improves on $\mathcal{T}_0$). Overall, thus, AOS-U with BERT-CTC PLs outperforms all other UOCL methods, and almost matches the performance of supervised ER, which, unlike AOS-U, can take advantage of both some stored past data and ground truth labels, and AOS.

6. Conclusion

We study and advance unsupervised OCL (UOCL) for ASR, enabling learning from non-i.i.d. streams of unlabeled data with minimal forgetting. Our exploration into UOCL reveals that it substantially mitigates forgetting, enabling models to closely align with supervised performance despite solely relying on unlabeled data. By implementing self-training techniques, we demonstrate the potential for ASR systems to continually adapt and learn from new, unlabeled tasks while preserving knowledge across old domains. Our findings highlight the robustness of CTC-based pseudo-labeling, focusing on acoustic evidence for consistent performance, and the BERT-CTC approach as an effective solution for adapting to text-domain shifts. Our proposed UOCL extensions outperform existing approaches, offering a robust solution for ASR systems in label-scarce environments. Our work paves the way for more resilient and adaptable ASR systems, capable of evolving in dynamic environments without the need for constant retraining with labeled data.

7. Acknowledgements

Research supported by Research Foundation Flanders (FWO) under grant S004923N of the SBO programme.

8. References

[1] M. McCloskey *et al.*, "Catastrophic interference in connectionist networks: The sequential learning problem," ser. Psychology of Learning and Motivation, 1989, vol. 24, pp. 109–165.

[2] H.-J. Chang *et al.*, "Towards Lifelong Learning of End-to-End ASR," in *Proc. Interspeech 2021*, 2021, pp. 2551–2555.

[3] S. Vander Eeckt and H. Van hamme, "Continual learning for monolingual end-to-end automatic speech recognition," in *2022 30th European Signal Processing Conference (EUSIPCO)*, 2022.

[4] ——, "Using adapters to overcome catastrophic forgetting in end-to-end automatic speech recognition," in *ICASSP 2023*, 2023.

[5] M. Sustek, S. Sadhu, and H. Hermansky, "Dealing with Unknowns in Continual Learning for End-to-end Automatic Speech Recognition," in *Proc. Interspeech 2022*, 2022, pp. 1046–1050.

[6] Y. Takashima *et al.*, "Updating only encoders prevents catastrophic forgetting of end-to-end ASR models," in *Proc. Interspeech 2022*. ISCA, 2022, pp. 2218–2222.

[7] G. Chennupati *et al.*, "Ilasr: Privacy-preserving incremental learning for automatic speech recognition at production scale," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, ser. KDD '22, 2022, p. 2780–2788.

[8] M. Yang *et al.*, "Online continual learning of end-to-end speech recognition models," in *Proc. Interspeech 2022*, 2022.

[9] S. Vander Eeckt and H. Van hamme, "Weight averaging: A simple yet effective method to overcome catastrophic forgetting in automatic speech recognition," in *ICASSP 2023*, 2023.

[10] ——, "Rehearsal-free online continual learning for automatic speech recognition," in *Proc. Interspeech 2023*, 2023.

[11] P. Buzzega *et al.*, "Dark experience for general continual learning: a strong, simple baseline," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 15 920–15 930.

[12] D. Madaan *et al.*, "Representational continuity for unsupervised continual learning," in *International Conference on Learning Representations*, 2022.

[13] N. Michel *et al.*, "Domain-aware augmentations for unsupervised online general continual learning," in *34th British Machine Vision Conference 2023*. BMVA, 2023.

[14] J. He and F. Zhu, "Unsupervised continual learning via pseudo labels," in *Continual Semi-Supervised Learning*. Cham: Springer International Publishing, 2022, pp. 15–32.

[15] J. Kahn, A. Lee, and A. Hannun, "Self-training for end-to-end speech recognition," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[16] T. Likhomanenko *et al.*, "slimIPL: Language-Model-Free Iterative Pseudo-Labeling," in *Proc. Interspeech 2021*, 2021.

[17] Y. Higuchi *et al.*, "Momentum Pseudo-Labeling for Semi-Supervised Speech Recognition," in *Proc. Interspeech 2021*, 2021, pp. 726–730.

[18] Y. Chen *et al.*, "Semi-Supervised ASR by End-to-End Self-Training," in *Proc. Interspeech 2020*, 2020, pp. 2787–2791.

[19] D. Hwang *et al.*, "Large-scale asr domain adaptation using self- and semi-supervised learning," in *ICASSP 2022*, 2022, pp. 6627–6631.

[20] Q. Xu *et al.*, "Iterative Pseudo-Labeling for Speech Recognition," in *Proc. Interspeech 2020*, 2020, pp. 1006–1010.

[21] ——, "Self-training and pre-training are complementary for speech recognition," in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[22] W. Hou *et al.*, "Cross-Domain Speech Recognition with Unsupervised Character-Level Distribution Matching," in *Proc. Interspeech 2021*, 2021, pp. 3425–3429.

[23] J. Zhou *et al.*, "Madi: Inter-domain matching and intra-domain discrimination for cross-domain speech recognition," in *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.

[24] H. Zhu *et al.*, "Boosting cross-domain speech recognition with self-supervision," in *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP), 2023*, 2023.

[25] G.-T. Lin *et al.*, "Listen, Adapt, Better WER: Source-free Single-utterance Test-time Adaptation for Automatic Speech Recognition," in *Proc. Interspeech 2022*, 2022, pp. 2198–2202.

[26] C. Kim, J. Park, H. Shim, and E. Yang, "SGEM: Test-time adaptation for automatic speech recognition via sequential-level generalized entropy minimization," in *Proc. Interspeech 2023*, 2023.

[27] S. Karita *et al.*, "Improving Transformer-Based End-to-End Speech Recognition with Connectionist Temporal Classification and Language Model Integration," in *Proc. Interspeech 2019*.

[28] N. Srivastava *et al.*, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

[29] Y. Higuchi *et al.*, "BERT meets CTC: New formulation of end-to-end speech recognition with pre-trained masked language model," in *Findings of the Association for Computational Linguistics: EMNLP 2022*, Dec. 2022, pp. 5486–5503.

[30] J. Devlin *et al.*, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*, Jun. 2019, pp. 4171–4186.

[31] D. Rolnick *et al.*, "Experience replay for continual learning," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[32] J. S. Vitter, "Random sampling with a reservoir," *ACM Trans. Math. Softw.*, vol. 11, no. 1, p. 37–57, mar 1985.

[33] S. Watanabe *et al.*, "ESPnet: End-to-end speech processing toolkit," in *Proceedings of Interspeech*, 2018, pp. 2207–2211.

[34] V. Panayotov *et al.*, "Librispeech: An asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[35] A. Mathur *et al.*, "Libri-adapt: a new speech dataset for unsupervised domain adaptation," *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7439–7443, 2020.

[36] F. Hernandez *et al.*, "Ted-lium 3: Twice as much data and corpus repartition for experiments on speaker adaptation," in *Speech and Computer*, 2018, pp. 198–208.

[37] R. Ardila *et al.*, "Common voice: A massively-multilingual speech corpus," in *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, 2020.

[38] A. Gulati *et al.*, "Conformer: Convolution-augmented Transformer for Speech Recognition," in *Proc. Interspeech 2020*, 2020, pp. 5036–5040.

[39] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[40] W.-N. Hsu *et al.*, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 29, p. 3451–3460, oct 2021.

[41] J. Kahn *et al.*, "Libri-light: A benchmark for asr with limited or no supervision," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7669–7673.

[42] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2018, pp. 66–71.

[43] D. S. Park *et al.*, "Specaugment: A simple data augmentation method for automatic speech recognition," *Interspeech 2019*, Sep 2019.

[44] H. Strik, C. Cucchiarini, and J. M. Kessens, "Comparing the recognition performance of csrs: in search of an adequate metric and statistical significance test," in *INTERSPEECH*, 2000.