Boosting Cross-Domain Speech Recognition with Self-Supervision

Han Zhu, *Student Member, IEEE,* Gaofeng Cheng, *Member, IEEE,* Jindong Wang, Wenxin Hou, Pengyuan Zhang, *Member, IEEE,* Yonghong Yan, *Member, IEEE,*

***Abstract*—The cross-domain performance of automatic speech recognition (ASR) could be severely hampered due to the mismatch between training and testing distributions. Since the target domain usually lacks labeled data, and domain shifts exist at acoustic and linguistic levels, it is challenging to perform unsupervised domain adaptation (UDA) for ASR. Previous work has shown that self-supervised learning (SSL) or pseudo-labeling (PL) is effective in UDA by exploiting the self-supervisions of unlabeled data. However, these self-supervisions also face performance degradation in mismatched domain distributions, which previous work fails to address. This work presents a systematic UDA framework to fully utilize the unlabeled data with self-supervision in the pre-training and fine-tuning paradigm. On the one hand, we apply continued pre-training and data replay techniques to mitigate the domain mismatch of the SSL pre-trained model. On the other hand, we propose a domain-adaptive fine-tuning approach based on the PL technique with three unique modifications: Firstly, we design a dual-branch PL method to decrease the sensitivity to the erroneous pseudo-labels; Secondly, we devise an uncertainty-aware confidence filtering strategy to improve pseudo-label correctness; Thirdly, we introduce a two-step PL approach to incorporate target domain linguistic knowledge, thus generating more accurate target domain pseudo-labels. Experimental results on various cross-domain scenarios demonstrate that the proposed approach effectively boosts the cross-domain performance and significantly outperforms previous approaches.**

***Index Terms*—Automatic Speech recognition, domain adaptation, self-supervised learning, pre-training, pseudo-labeling**

I. Introduction

THE performance of end-to-end (E2E) automatic speech recognition (ASR) systems has improved dramatically over the past years [1]–[5] due to the advanced neural network architectures, improved training criteria, and large amounts of training data. However, the performance degradation on the cross-domain data is still a challenging issue for ASR due to the domain shift between the training and testing data. Since it is impossible to cover all test domains in the training data, applying domain adaptation [6] for a new target domain is of great interest in the application of ASR.

Domain adaptation aims to transfer a model trained on the source data to a given target domain in supervised or unsupervised conditions. When the labeled data is available in the target domain, the supervised domain adaptation is straightforward since we could simply use the labeled data to fine-tune the source model [7], [8]. However, since the labeled target data is costly to collect, the unsupervised domain adaptation (UDA) scenario, where no labeled data is available in the target domain, is more desired in real-world applications.

Existing UDA approaches tackle the lack of labeled data issue from different aspects. An intuitive solution is to synthesize target domain data [9]–[14]. However, these approaches require a specific design for the target domain [9] or careful tuning of the data generation model [11], making them inconvenient when extending to an arbitrary new domain or large-scale applications. Another category is domain-invariant feature learning with distribution matching approaches [15], [16], which aims to learn a domain-invariant representation while being class-discriminative on the source domain. However, the class-discriminative representations on the target domain cannot be easily guaranteed. Thus, it would fail under certain domain mismatch scenarios [17]–[19].

Recently, self-supervised learning (SSL) based pre-training [20]–[23] and pseudo-labeling (PL) [24], [25] are shown to be effective for UDA of the E2E-ASR model by directly training on the unlabeled target data with self-supervision. On the one hand, both SSL and PL approaches are simple in practice and do not need customization for specific domains. Thus, they are convenient to be applied to any new domain and large-scale applications. On the other hand, SSL and PL are shown to be robust under various domain mismatch conditions [23]–[25].

Nonetheless, existing literature typically focused on one aspect to address the UDA problem, e.g., with SSL [23], online [25] or offline PL [24]. This practice failed to realize the full potential of self-supervision for UDA. To push the limits of UDA, in this work, we first identify the weaknesses of SSL and PL when applying them to the UDA scenario and propose innovative solutions to address them. Then, we seamlessly integrate the improved SSL and PL approaches into a systematic UDA framework to boost **C**ross-dom**A**in **S**peech recogni**T**ion with Se**L**f-Sup**E**rvision, namely, **CASTLE**.

In summary, the major novelties of this paper are as follows:

- *Dual-Branch PL (DPL):* There are two vital challenges in existing online PL approaches: Firstly, since the self-generated pseudo-labels are used as the supervision, the errors would be accumulated and cause the error accumulation [26] (or the confirmation bias [27]) issue, degrading the performance and sometimes driving the training to

H. Zhu, G. Cheng, P. Zhang and Y. Yan are with the Key Laboratory of Speech Acoustics and Content Understanding, Institute of Acoustics, Chinese Academy of Sciences, Beijing, China, and also with the University of Chinese Academy of Sciences, Beijing, China (e-mail:zhuhan@hccl.ioa.ac.cn;chenggaofeng@hccl.ioa.ac.cn ;zhangpengyuan@hccl.ioa.ac.cn;yanyonghong@hccl.ioa.ac.cn).

J. Wang (jindong.wang@microsoft.com) is with Microsoft Research Asia, China.

W. Hou (wenxinhou@microsoft.com) is with Microsoft STCA, China.

collapse. Secondly, due to the lack of confidence filtering design for online PL, all pseudo-labels are used in training, including the extremely noisy pseudo-labels. To address these challenges, on the one hand, DPL proposes to break the error accumulation chain by using an auxiliary branch to generate pseudo-labels. On the other hand, DPL utilizes a specifically designed confidence estimation that discards the CTC blank scores, thus robustly improving the online PL. Extensive experiments demonstrated the advantage of DPL over existing online PL approaches.

- *Uncertainty-Aware Confidence Filtering (UCF):* Most existing filtering methods for offline PL utilize decoding scores as confidence estimation to rule out the noisy pseudo-labels [28], [29]. However, this confidence estimation is unreliable when ASR networks are poorly calibrated [30]. UCF addressed this issue by adaptively utilizing uncertainty [31] and confidence estimations to select pseudo-labels in offline PL, where the combination hyper-parameters in UCF are adaptively determined on the development set. Experiments demonstrated that UCF can outperform existing filtering approaches without tuning hyper-parameters.
- *Two-Step PL:* Two-step PL is an empirically motivated approach that utilizes the offline PL to refine the online PL linguistically. Although a simple combination of two types of PL approaches, it consistently outperforms either one of them in practice.
- *Continued Pre-Training with Data Replay:* Catastrophic forgetting is a critical issue in continual learning. Since continued pre-training [23], [32] is proven effective and used in our approach, we examine whether catastrophic forgetting is severe here by evaluating the effectiveness of data replay [33], which is widely adopted to address knowledge forgetting. Based on experimental findings, we provide suggestions on continued pre-training strategies given different fine-tuning strategies.

We performed detailed experiments on various cross-domain datasets and showed that the proposed approach CASTLE could effectively boost the cross-domain performance of ASR and consistently outperforms previous UDA approaches.

The rest of the paper is organized as follows. In Section II we formulate the UDA problem and review related works. Then the proposed approach CASTLE is introduced in Section III. We describe experimental settings in Section IV, and then present experimental results in Section V. Finally, Section VII concludes the paper.

II. Preliminaries and Related Work

We first define the notations. In an ASR model, the acoustic feature $\mathbf{X}$ is first processed by a feature transformation function $g : \mathcal{X} \mapsto \mathcal{Z}$ to generate the latent feature representation $\mathbf{Z}$. Then the projection function $h : \mathcal{Z} \mapsto \mathcal{P}$ gives the final prediction of the ASR network $\mathbf{P}$. Finally, the decoding function $d : \mathcal{P} \mapsto \mathcal{Y}$ generates the transcription $\mathbf{Y}$. The composite transformation of the ASR network is $f = g \circ h : \mathcal{X} \mapsto \mathcal{P}$. And f, g, h are parameterized by θ, ϕ, ψ respectively.

Then we formulate the UDA problem. Suppose there are two different domains: source domain S and target domain T. In source domain, there is a large unlabeled dataset $\mathbb{U}^S = \left\{\mathbf{X}_1^S, \ldots, \mathbf{X}_N^S\right\}$ and a small labeled subset $\mathbb{L}^S = \left\{\left(\mathbf{X}_1^S, \boldsymbol{Y}_1^S\right), \ldots, \left(\mathbf{X}_M^S, \boldsymbol{Y}_M^S\right)\right\}$, where $M \leq N$ and $(\mathbf{X}, \boldsymbol{Y})$ denote an feature-transcription pair. In target domain, only an unlabeled dataset $\mathbb{U}^T = \left\{\mathbf{X}_1^T, \ldots, \mathbf{X}_O^T\right\}$ is available. The goal of UDA is to improve the ASR performance on the target domain using all above datasets. In this work, we also assume the target domain style text corpus is available to train a target domain LM, as it is easy to collect. Moreover, a development set, i.e., a small labeled source domain dataset, is used during the training process.

The major challenge of the UDA scenario is the lack of labeled target data so that we can't directly perform supervised training on the target domain. In the following, we introduce mainstream UDA approaches for ASR, which utilize unlabeled target data in different ways. Briefly, domain-invariant feature learning uses a distribution match loss to match the distribution between labeled and unlabeled data. SSL-based approach exploits a self-supervised loss to learn a better representation for target domain distribution. PL-based approaches generate pseudo-labels for unlabeled data and then train the model on it.

A. Domain-Invariant Feature Learning

Learning domain-invariant features with distribution matching is a prominent UDA approach in many areas, which aims to generalize better to the target domain by minimizing the differences between the intermediate feature representations of the source and target domain. The distribution matching loss could be formulated as $D\left(g_\phi(X^S) \| g_\phi(X^T)\right)$, where D is the distance metric. Specifically, some works [15] used Maximum Mean Discrepancy to measure and reduce the differences between two domains. Others [16], [34] chose domain adversarial training [35] to learn a domain-invariant representation to fool the domain classifier, which can be viewed as minimizing the Jensen-Shannon (JS) divergence [36]. However, such practice is unreliable when label distribution shift [17], conditional distribution shift [18], or large domain discrepancy [19] exist.

B. Self-Supervised Learning

Since SSL-based pre-training is shown to be effective in exploiting unlabeled data for ASR [37]–[46], it is suitable to use SSL to utilize unlabeled source and target data for UDA [23].

Robust wav2vec 2.0 [23] showed that joint pre-training on unlabeled target data could improve target domain performance when fine-tuning on the labeled source data. However, since joint pre-training is time-consuming [23], thus continued pre-training [32] would be more appealing with low computation budget. Nonetheless, continued pre-training may suffer from the catastrophic forgetting of source knowledge [33], which we study in detail in this work.

Joint supervised and self-supervised training [47]–[49] was shown to be better than self-supervised pre-training in settings with domain mismatch. However, in the studied UDA scenario,

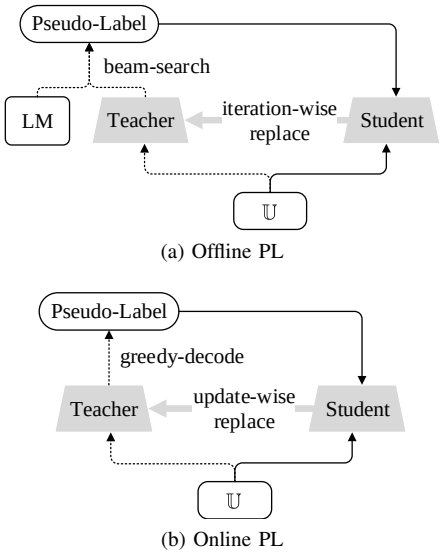


Fig. 1. Illustration of generating pseudo-labels and updating the teacher model in offline and online PL. The dashed line denotes inference mode and the solid line denotes training mode[1].

Algorithm 1 CASTLE algorithm.

Input: Labeled source dataset $\mathbb{L}^S$, unlabeled target dataset $\mathbb{U}^T$, SSL model $\mathcal{M}_{\text{SSL}}$ pre-trained on $\mathbb{U}^S$, target domain LM $\mathcal{M}_{\text{LM}}$, hyper-parameters α, K, c_{on}.
Output: ASR model $\mathcal{M}_{\text{ASR}}$.

Continued pre-train $\mathcal{M}_{\text{SSL}}$ on $\mathbb{U}^T$;
Add random initialized linear layer on $\mathcal{M}_{\text{SSL}}$ to produce $\mathcal{M}_{\text{ASR}}$;
repeat
  Draw batches $\mathbf{B}^S$, $\mathbf{B}^T$ of the same size from $\mathbb{L}^S$ and $\mathbb{U}^T$;
  Filter $\mathbf{B}^T$ with the confidence score to produce $\mathbf{B}^{T\prime}$;
  Compute DPL loss $\mathcal{L}_{\text{DPL}}$ with $\mathbf{B}^S$ and $\mathbf{B}^{T\prime}$;
  Update the model $\mathcal{M}_{\text{ASR}}$ with $\mathcal{L}_{\text{DPL}}$;
until maximum updates for online PL are reached
repeat
  Decode $\mathbb{U}^T$ with $\mathcal{M}_{\text{ASR}}$ and $\mathcal{M}_{\text{LM}}$ to get $\hat{\mathbb{L}}^T$;
  Filter $\hat{\mathbb{L}}^T$ with UCF to produce $\hat{\mathbb{L}}^{T\prime}$;
  repeat
    Draw a batch $\hat{\mathbf{B}}^T$ from $\hat{\mathbb{L}}^{T\prime}$;
    Compute offline PL loss $\mathcal{L}_{\text{offlinePL}}$ with $\hat{\mathbf{B}}^T$;
    Update the model $\mathcal{M}_{\text{ASR}}$ with $\mathcal{L}_{\text{offlinePL}}$;
  until maximum updates for current iteration are reached
until maximum iterations for offline PL are reached

we failed to achieve better performance than the pure self-supervised approach with such joint training method, specifically, applying wav2vec 2.0 loss on unlabeled target data and CTC loss on labeled source data. One possible reason is that wav2vec 2.0 and CTC encourage different representation patterns [50]. Minimizing two losses alternatively with separate optimizers [48], or using some separate layers for each loss [49] can possibly address this issue. We stick to the self-supervised pre-training as it is more easily implemented and leave these explorations in the future.

C. Pseudo-labeling

Pseudo-labeling (PL) [51]–[53], also known as self-training, is widely used for semi-supervised learning due to its effectiveness and simplicity.

There are two mainstream implementations of PL in ASR: offline PL [54]–[58] and online PL [59], [60]. Both approaches use a teacher to generate pseudo-labels and a student to utilize pseudo-labels. The student is trained with back-propagation while the teacher is replaced with the student after certain intervals, which means the teacher and student have the same structure.

Online and offline PL differ in generating pseudo-labels and updating the teacher model. We illustrate the differences in Fig. 1. In online PL, pseudo-labels are generated online in each training update. In offline PL, pseudo-labels are generated offline before each training iteration[2]. When we formulate online/offline PL in the teacher-student framework, the fundamental difference between online and offline PL is the update frequency of the teacher model: online and offline PL update the teacher model after each training update and each training iteration, respectively.

[1]Dropout and data augmentations are enabled in training mode but disabled in inference mode.

[2]Each training iteration consists of multiple training updates.

Conventionally, offline PL is used as an LM-based PL approach where the LM is integrated when generating pseudo-labels [29], [58], [61]–[63]. On the contrary, Since the online generation of pseudo-labels requires the decoding speed to be fast, online PL approaches usually apply greedy-decoding, thus being an LM-free PL approach. Despite its simplicity, online PL could achieve competitive performance compared with offline PL [60].

Since pseudo-labels are noisy, filtering techniques could be applied to select pseudo-labels with better quality. Confidence filtering approaches [28], [29] utilize the decoding score to select pseudo-labels. However, the poorly calibrated neural networks could produce over-confident erroneous predictions and make the confidence filtering unreliable [30]. To improve the confidence estimation, some model-based approaches utilize an additional confidence estimation module [30]. But they also complicate the training procedure. As an alternative, some work [24] used the uncertainty estimation of the ASR model to filter out the erroneous pseudo-labels. The uncertainty could be modeled with the prediction variance [64] and estimated via the Monte Carlo dropout [31]. Apart from the filtering techniques, prediction combination [65], [66] can also be used to improve the quality of pseudo-labels.

In terms of the UDA scenario for ASR, offline PL is shown to be effective in [21], [24], where [24] applied the dropout-based uncertainty filtering and [21] utilized the model-based confidence filtering [30]. Online PL was explored in MPL [25], which utilizes the EMA technique to stabilize training.

III. Proposed Approach

The proposed CASTLE approach consists of the following stages. Given a pre-trained SSL model, e.g., wav2vec 2.0, we first alleviate the pre-training mismatch by continued pre-training on the unlabeled target domain $\mathbb{U}^T$. After that, we perform the domain-adaptive fine-tuning to resolve the fine-tuning mismatch. Specifically, we utilize the two-step PL to conduct the online and offline PL step by step, where the

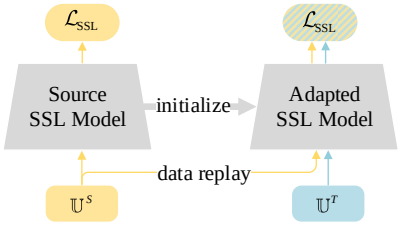


Fig. 2. Illustration of continued pre-training with data replay.

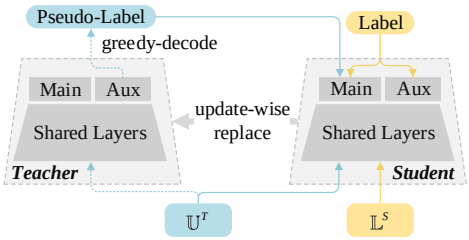


Fig. 3. Illustration of dual-branch learning in DPL.

dual-branch PL is used as the online PL approach and the uncertainty-aware confidence filtering is used during offline PL. In this work, we concentrate on CTC [67] models, which enable non-autoregressive decoding and achieve state-of-the-art results for many low-resource scenarios [37], [68]–[70]. We summarize the complete training procedure in Alg. 1. And the details are described as follows.

A. Continued Pre-Training with Data Replay

Continued pre-training on the target domain could effectively alleviate the domain mismatch of the SSL pre-training model while remaining light-weight computation [23]. However, the continued pre-training could lead to the catastrophic forgetting of the source knowledge because the model only observes the target data during this stage. Suppose the final SSL model is only fine-tuned on the labeled source data, and we expect generalization to the target domain. In that case, it is preferable that both source and target knowledge are well aligned in the SSL model.

To resolve this issue, as shown in Fig. 2, we could replay the source data during continued pre-training. Specifically, we select source and target data with a ratio $p_{s/t}$. The ratio $p_{s/t}$ is smaller than 1 since the source model is already well trained on the source data. And the source data is only used as the regularization to avoid knowledge forgetting.

B. Online PL with Dual-Branch PL

In the online PL approach, since pseudo-labels are generated in each update, the error accumulation issue is much more severe. Dual-branch PL (DPL) could effectively alleviate this issue. We explain the details of DPL as follows.

1) Dual-Branch Learning: The main idea of DPL is the dual-branch learning, which utilizes partially separated parameters to generate and exploit pseudo-labels, thus breaking the error accumulation chain. We illustrate the training procedure of dual-branch learning in Fig. 3 and explain it as follows.

In dual-branch learning, the model consists of a shared feature transformation function g_ϕ, and two projection functions h_{ψ_a} and h_{ψ_m}, where h_{ψ_a} is the auxiliary branch and h_{ψ_m} is the main branch. These two branches are stacked on top of g_ϕ. Note that the projection function h corresponds to the final linear layer of the CTC model in this work.

For samples in the unlabeled target dataset $\mathbb{U}^T$, the model generates pseudo-labels from the auxiliary branch:

$$\hat{\mathbf{Y}} = \text{greedy-decode}\left(\widetilde{h}_{\psi_a}(\widetilde{g}_\phi(\mathbf{X}))\right), \mathbf{X} \in \mathbb{U}^T, \tag{1}$$

where $\widetilde{h}$ and $\widetilde{g}$ are h and g in the inference mode.

Then the online PL loss is computed between predictions from the main branch and the above pseudo-labels:

$$\mathcal{L}_{\text{onlinePL}} = \text{CTC}\left(h_{\psi_m}(g_\phi(\mathbf{X}')), \hat{\mathbf{Y}}\right), \mathbf{X} \in \mathbb{U}^T, \tag{2}$$

where $\mathbf{X}'$ is the augmented version of $\mathbf{X}$.

For samples in the labeled source dataset $\mathbb{L}^S$, both the auxiliary branch and the main branch are trained on them as:

$$\begin{aligned}\mathcal{L}_{\text{SUP}} = &\text{CTC}\left(h_{\psi_a}(g_\phi(\mathbf{X}')), \mathbf{Y}\right) \\ &+ \text{CTC}\left(h_{\psi_m}(g_\phi(\mathbf{X}')), \mathbf{Y}\right), (\mathbf{X}, \mathbf{Y}) \in \mathbb{L}^S,\end{aligned} \tag{3}$$

Finally, the total loss is:

$$\mathcal{L}_{\text{DPL}} = \mathcal{L}_{\text{SUP}} + \alpha \mathcal{L}_{\text{onlinePL}}, \tag{4}$$

where α is a hyper-parameter to be tuned.

In dual-branch learning, both labeled source data and unlabeled target data is used to optimize the main branch, while only labeled source data is used to optimize the auxiliary branch. Since pseudo-labels are generated with the auxiliary branch while exploited by the main branch, DPL could break the error accumulation chain. And since the shared feature transformation function is optimized by both source and target data, the latent feature representation could generalize to both domains. Thus the auxiliary branch that is only optimized with source data could also produce accurate enough pseudo-labels on the target domain. Moreover, the auxiliary branch will be discarded after training, thus keeping the same inference computation.

2) One Stage Training with Confidence Filtering: Previous online PL approaches usually consist of supervised training as the first stage and PL training as the second stage. This practice could avoid training on noisy pseudo-labels in the beginning but neglects the fact that pseudo-labels could still be noisy in the second stage. This issue is especially serious with domain mismatches.

To resolve this issue, we use the confidence filtering to automatically choose the pseudo-labels since the confidence score could roughly reflect the quality of pseudo-labels.

Although there are many approaches to estimate the confidence score for offline PL [28], [29], few attempts have been made for online PL. In offline PL, pseudo-labels are mostly generated with beam-search, which naturally generates a decoding score that can be used as the confidence estimation. On the other hand, greedy search is used in online PL, and the decoding score for greedy search is not properly defined.

To fill this gap, we design a confidence estimation for online PL. Specifically, we compute the confidence score by taking the mean value of the maximum scores in all frames, which shares a similar idea with the decoding score in offline PL. Our unique design is: to avoid the trivial solution that the model always predicts blank tokens in CTC [60], [69], we discard the frames where the maximum score belongs to the blank token. A similar idea is used in [71] to improve the confidence measure of the CTC-based ASR model.

Formally, given an sample $\mathbf{X} \in \mathbb{U}^T$, the prediction of the ASR network is computed as:

$$\mathbf{P} = \text{log-softmax}\left(f_\theta\left(\mathbf{X}\right)\right), \mathbf{X} \in \mathbb{U}^T, \tag{5}$$

where $\mathbf{P} \in \mathbb{R}^{T \times V}$. T is the number of frames and V is the output vocabulary size including the blank token.

Then we remove the frames $\mathbf{P_t}$ where the maximum score belongs to the blank token. And the resulting prediction is $\mathbf{P'} \in \mathbb{R}^{T' \times V}$.

Finally, the confidence score is estimated as:

$$C_{\text{on}} = \exp\left(\frac{1}{T'}\sum_{t=1}^{T'}\max_{v}\left(\mathbf{P'}_{t,v}\right)\right) \tag{6}$$

It is worth noting that this confidence estimation does not require extra parameters or computations, thus fitting in the online PL approach.

To this end, the PL loss $\mathcal{L}_{\text{onlinePL}}$ will only include samples with confidence $C_{\text{on}} \geq c_{\text{on}}$, where c_{on} is the filtering threshold.

3) Sampling Strategy: We utilize a fixed sampling strategy where the labeled source data and unlabeled target data are sampled with the same batch size in each update. In this way, a stable supervised signal exists in each update to regularize the PL training. And we could control the weight of the unlabeled data with the PL loss weight α in Eq. (4) . A proper α value could effectively stabilize training without using EMA, which is shown to be important for stabilization if we randomly sample from the mixed dataset of both labeled and unlabeled data [25] or only sample the unlabeled data [60]. Therefore, we could avoid maintaining an additional EMA version of the model, thus being more memory efficient.

C. Offline PL with Uncertainty-Aware Confidence Filtering

Similar to online PL, offline PL generates pseudo-labels for the unlabeled target data with the teacher model as:

$$\hat{\mathbf{Y}} = \text{beam-search}\left(\widetilde{f_\xi}(\mathbf{X}), \mathcal{M}_{\text{LM}}\right), \mathbf{X} \in \mathbb{U}^T, \tag{7}$$

where ξ is the parameter of the teacher model, which is replaced with the up-to-date student model θ after each iteration of offline PL. $\widetilde{f}$ is f in the inference mode. And $\mathcal{M}_{\text{LM}}$ is the target domain LM.

Then, the offline PL loss is computed as:

$$\mathcal{L}_{\text{offlinePL}} = \text{CTC}\left(f_\theta(\mathbf{X'}), \hat{\mathbf{Y}}\right), \mathbf{X} \in \mathbb{U}^T, \tag{8}$$

where θ is parameter of the student model and $\mathbf{X'}$ is the augmented version of $\mathbf{X}$.

[3]Unlike previous notations, the training mode here enables dropout but disables data augmentation.

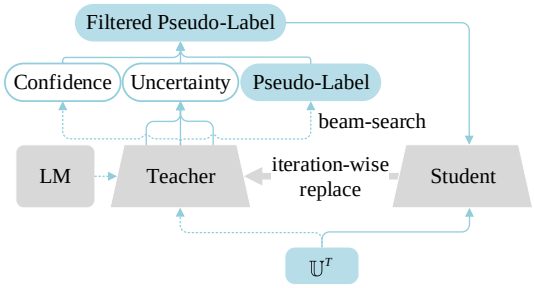


Fig. 4. Illustration of offline PL with uncertainty-aware confidence filtering (UCF). Pseudo-labels and confidence scores are generated in the inference mode (dashed line). And the uncertainty estimation is generated in the training mode[3] (solid line).

To improve the quality of offline pseudo labels $\hat{\mathbf{Y}}$, we could adopt filtering approaches based on confidence or uncertainty estimations. However, the confidence score is unreliable for a poorly calibrated neural network and we find low uncertainty alone also can't guarantee the accuracy of selected pseudo-labels. Intuitively, pseudo-labels with higher confidence and low uncertainty are more likely to be correct. Therefore, we integrate the confidence and uncertainty in one framework for filtering, i.e., uncertainty-aware confidence filtering (UCF). We illustrate UCF in Fig. 4 and describe the details as follows.

The decoding score from the beam-search decoding in Eq. (7) could be used as the confidence score C_{off}. And the C_{off} is normalized by the sample length.

As for the uncertainty, we adopt Monte Carlo dropout [31] to estimate the prediction variance with some modifications for the ASR task. The procedure is described below.

Firstly, we compute pseudo-labels with dropout enabled for K times:

$$\hat{\mathbf{Y}}_k = \text{beam-search}\left(f_\xi(\mathbf{X}), \mathcal{M}_{\text{LM}}\right), \mathbf{X} \in \mathbb{U}^T, \tag{9}$$

Then, the uncertainty could be estimated with the prediction variance as:

$$U = \frac{1}{K|\hat{\mathbf{Y}}|}\sum_{k=1}^{K} D\left(\hat{\mathbf{Y}}_k \| E(\hat{\mathbf{Y}}_k)\right) \tag{10}$$

where D is the distance metric. Since $\mathbf{Y}$ is the text, we utilize the edit distance (Levenshtein distance) here.

However, it is unknown how to compute the mean text $E(\hat{\mathbf{Y}})$. Since the decoding results with dropout could be viewed as the deviations of the one without dropout, we could use the approximation that $\hat{\mathbf{Y}} = E(\hat{\mathbf{Y}}_k)$, which means using the decoding result without dropout as the mean text.

To this end, the uncertainty could be estimate with:

$$U = \frac{1}{K|\hat{\mathbf{Y}}|}\sum_{k=1}^{K} \text{edit-distance}\left(\hat{\mathbf{Y}}_k \| \hat{\mathbf{Y}}\right) \tag{11}$$

Compared with the uncertainty estimation in previous work [24], our estimation uses the mean of edit distances instead of the maximum since it is more mathematically intuitive.

Moreover, we find the model tends to select shorter samples in some cases. But those selected shorter samples are still erroneous. Therefore, we add a length bonus term $log(|\hat{\mathbf{Y}}|)$ to alleviate this issue:

Pseudo-Labels w/ LM		
Substitution: 0 Deletion: 1 Insertion: 0	Reference	what's to be done at this point increase that amount for sometime so IT will eventually go down or i mean i imagine if you're just servicing the debt you're not really making any progress right
	Hypothesis	what's to be done at this point increase that amount for sometime so ** will eventually go down or i mean i imagine if you're just servicing the debt you're not really making any progress right
Pseudo-Labels w/o LM		
Substitution: 5 Deletion: 1 Insertion: 0	Reference	what's to be done at this point increase that amount for sometime so IT WILL eventually go down or i MEAN I imagine if you're just servicing the DEBT you're not REALLY making any progress right
	Hypothesis	what's to be done at this point increase that amount for sometime so ** WIL eventually go down or i MAN II imagine if you're just servicing the DET you're not REALLYE making any progress right

Fig. 5. Illustration of errors in pseudo-labels generated w/ or w/o LM. The same CTC-based ASR model is used to generate these pseudo-labels.

Finally, the UCF strategy would select samples with:

$$C_{\text{off}} - \gamma U + \eta \log(|\hat{\mathbf{Y}}|) >= c_{\text{off}} \tag{12}$$

where c_{off} is the filtering threshold. γ and η are hyper-parameters that can be directly estimated on the development set, thus alleviating the effort for hyper-parameter tuning.

The estimation of the uncertainty requires multiple forward computations for each sample. For offline PL, since the estimation is only conducted after each iteration, the increased computation is tolerable. But it will significantly increase the training time for online PL since the estimation is conducted in each update. Therefore, we only apply UCF for offline PL.

D. Two-Step PL

Benefiting from the continuous improvement of pseudo-labels [60], the ASR model could be fast adapted to the target domain with online PL. However, online PL purely relies on the source domain transcripts to learn linguistic knowledge, while the target domain linguistics are neglected. Consequently, pseudo-labels generated in online PL are likely to be incorrect in linguistics. We illustrate this effect in Fig. 5. After the adaptation with online PL, we use the adapted model to generate pseudo-labels with or without LM. We take an utterance from the SWBD training set as an example. The ground-truth labels are denoted as the reference and pseudo-labels are denoted as the hypothesis. When LM is used, there is only one deletion error. However, when decoding without LM, there are five additional substitution errors, roughly correct in acoustic but incorrect in linguistics.

Training on these erroneous pseudo-labels would be sub-optimal. Thus, we could consider using the target domain LM to improve the accuracy of pseudo-labels. Although there are no transcripts for the target domain samples, it is not that hard to collect some text with the target domain style. Target domain LM could be naturally integrated into offline PL by generating pseudo-labels with the beam-search decoding and LM. If the target domain LM is used, offline PL could inject the linguistic knowledge of the target domain into the ASR model by training on these pseudo-labels. To this end, we propose the two-step PL strategy that can be summarized as: online PL for on-the-fly adaptation followed by offline PL for further linguistic refinement. In comparison to pure online PL, two-step PL addresses linguistically incorrect pseudo-labels with the refinement of offline PL that relies on LM-based decoding. Compared with pure offline PL, two-step PL utilizes online PL to offer a superior seed model for offline PL, thereby improving final performance.

IV. Experimental Setup

A. Corpus

1) Source Domain Corpus: The source domain pre-trained model is the wav2vec 2.0 base model pre-trained on Librispeech 960h[4]. Consequently, LibriSpeech [72] is used as the source domain dataset. We use the entire 960h training set as the unlabeled source domain dataset and the 100h subset as the labeled source domain dataset.

2) Target Domain Corpus: We use several cross-domain corpora to evaluate the performance under different mismatched situations. Specifically, we use Indian English Common Voice (CV), TED-LIUM v3 (TED) [73] and SwitchBoard (SWBD) [74] as unlabeled target domain datasets. Indian English Common Voice (CV) is extracted from the original Common Voice corpus [75] by selecting Indian accented data. We split the training, development and testing sets with the proportion of 8 : 1 : 1, respectively. For TED, standard train/dev/test splits are used. In terms of SWBD, the development set is RT-03S [76] and the testing sets are Hub05 Eval2000 [77] SwitchBoard (H-SB) and CallHome (H-CH) subsets. All audios are re-sampled to 16kHz and transcripts are pre-processed to upper-case letters and no punctuation except apostrophes. Therefore, the corpora in source and target domain have the same speech and transcription formats. The details of the target domain datasets are shown in Table I.

TABLE I
THE STRUCTURE OF TARGET DOMAIN CORPUS (HOURS).

Split	CV	TED	SWBD
Train	48	452	319
Dev	6	2	6
Test	6	3	4

3) Corpus for Target Domain LM: Since the offline PL always requires decoding the training set with the target domain LM, the text corpus for LM should not contain the transcripts of the training set. For CV, we use the LM of Librispeech as the target domain LM since they are both read

[4]The pre-trained model is downloaded from the wav2vec 2.0 repository: https://github.com/pytorch/fairseq/blob/master/examples/wav2vec

TABLE II
COMPARISON WITH UNSUPERVISED BASELINES AND SUPERVISED TOPLINES.

Method	Pre-Train Data		Fine-Tune Data			WER%						
	Source	Target	Source	Target		CV		TED		SWBD		
	Unlabeled	Unlabeled	Labeled	Labeled	Unlabeled	Dev	Test	Dev	Test	RT03	H-SB	H-CH
Unsupervised results												
Wav2vec 2.0	960h	×	100h	×	×	35.7	36.2	10.2	10.6	35.9	25.8	34.1
Robust wav2vec	960h	all	100h	×	×	22.2	22.4	8.8	8.7	24.8	17.2	23.9
+ DAT	960h	all	100h	×	all	19.5	19.6	8.9	8.6	24.6	17.2	23.9
+ MPL	960h	all	100h	×	all	16.4	16.4	7.6	7.2	18.4	12.9	18.3
+ DUST	960h	all	100h	×	all	18.5	18.3	7.3	6.9	18.5	12.2	18.8
CASTLE (online PL)	960h	all	100h	×	all	**15.8**	**15.9**	6.8	6.5	17.0	11.8	17.6
CASTLE (offline PL)	960h	all	100h	×	all	17.7	17.6	7.2	6.9	17.3	11.2	17.6
CASTLE (two-step PL)	960h	all	100h	×	all	**15.8**	**15.9**	**6.6**	**6.3**	**16.1**	**10.8**	**17.1**
Supervised results												
Wav2vec 2.0	960h	×	×	all	×	13.8	13.9	7.1	7.0	12.8	7.6	13.5
Robust wav2vec	960h	all	×	all	×	13.7	13.6	7.0	6.8	12.4	**7.1**	13.3
	960h	all	100h	all	×	**13.5**	**13.3**	**6.4**	**6.7**	**12.3**	7.4	**13.2**

speech. For TED, we use the official LM training corpus of TED [78], which is extracted from publicly available corpora distributed within the WMT 2013 machine translation evaluation campaign and is not overlapped with the training transcripts of TED. In terms of SWBD, we use transcripts of Fisher [79] dataset since they are both conversational speech.

B. Implementation Details

All experiments are conducted using the fairseq [80] toolkit. The code is publicly available[5].

For continued pre-training in CASTLE and other compared approaches, we use the effective batch size of 89.6m samples. The total training updates are 20k. And the learning rate is decayed from 5×10^{-5} without warmup. Continued pre-training applies the same time dimensional masking strategy with the source domain pre-training [37].

As for fine-tuning in CASTLE and other compared approaches, the effective batch size is 51.2m samples. The maximum learning rate is 3×10^{-5} and the tri-state learning rate schedule [37] is adopted. The convolution feature encoder is fixed during fine-tuning. The masking strategy during fine-tuning is masking in both time and channel dimensions, similar to SpecAugment [81].

In CASTLE, the training updates for online PL is 20k. The offline PL consists of two iterations, where each iteration has 5k updates. There are various hyper-parameters in CASTLE: DPL loss weight α, online filtering threshold c_{on}, offline filtering threshold c_{off}, UCF's hyper-parameters γ, η and K. Nonetheless, we find in practice that we can fix most hyper-parameters and automatically determine the others on the development set. Therefore, we do not need to tune them for a given dataset. Specifically, we set $\alpha = 1$, $c_{\text{on}} = 0.8$, $K = 3$, c_{off} is set to a value that 50% of development set's pseudo-labels would be selected, γ and η are set by minimizing the WER of the selected 50% of the development set's pseudo-labels. Note that fixing most or all hyper-parameters is widely adopted in existing approaches [51], [65], [82].

[5]https://github.com/zhu-han/CASTLE

As for evaluation, beam-search decoding with the 4-gram target domain LM is used to evaluate the performance.

Note that the two-step PL approach is more complicated than the one-step PL approach. A simpler method would be more desirable in certain situations, such as large-scale ASR training. Hence, in addition to the two-step PL version of CASTLE, we introduce one-step PL versions of CASTLE that solely employ either online PL or offline PL. These three versions of CASTLE offer a trade-off between performance and complexity, allowing users to select a variation according to their preferences.

V. RESULTS

A. Comparison with Previous Approaches

We implement UDA approaches in previous literature and compare them with the proposed approach CASTLE. Since most previous approaches tackle only pre-training or fine-tuning mismatch, we also implement their combinations to formulate stronger baselines that address both pre-training and fine-tuning mismatch. Specifically, we compare with:

- *Wav2vec 2.0*: We directly fine-tune the source domain wav2vec 2.0 model with labeled source data for a maximum of 30k updates.
- *Robust Wav2vec*: We follow the efficient implementation in robust wav2vec 2.0 [23], i.e., continued pre-training on the target domain. Training updates for continued pre-training and fine-tuning are 20k and 30k respectively. Since the robust wav2vec 2.0 is consistently better than wav2vec 2.0, we use it as the pre-trained models in the following three approaches.
- *DAT*: Domain adversarial training (DAT) [16] is used to learn domain-invariant features. The model is simultaneously optimized with the supervised loss on the source domain and the DAT loss between the source and target domain for 30k updates. The weight of the DAT loss is tuned on the development sets.
- *MPL*: Momentum pseudo-labeling (MPL) [25] utilizes the EMA model to generate pseudo-labels in online PL, thus being more stable. The model is first trained on labeled

source data for 10k updates, then trained with all data for another 20k updates. In the second stage, we randomly sample each batch from the mixed dataset of the source and target data as in [25]. The discount factor of EMA is tuned on the development sets.

- *DUST*: Uncertainty-driven self-training (DUST) [24] applies the prediction uncertainty to filter pseudo-labels during offline PL. The model is first trained on labeled source data for 10k updates. Then, we perform offline PL on target data for 4 iterations and each iteration consists of 5k updates. The target domain LM is used to generate pseudo-labels. To get the uncertainty estimation, we compute pseudo-labels with dropout for 3 times.

As shown in the upper part of Table II, by leveraging additional unlabeled target domain data with the continued pre-training technique, the robust wav2vec model consistently outperforms the wav2vec 2.0 model on all datasets.

On the basis of the robust wav2vec model, DAT could further improve the performance on CV datasets by encouraging the invariant representation between source and target domain. However, such improvement could not generalize to TED and SWBD datasets. The reason is that the CV dataset and the source domain dataset are both read speech, while TED and SWBD are in different linguistic domains, i.e., lecture and conversational speech. Although DAT is observed to be effective for ASR in previous work, they mostly use it under the acoustic domain shift, e.g., mismatches of environment, device, accent, etc. And the linguistic style of the source and target domain are similar, e.g., both read speech. This phenomenon is in line with the findings in [17], which indicates DAT is less effective with the label distribution shift.

On the contrary, both online (MPL) and offline (DUST) PL approaches could effectively boost the cross-domain performance on all datasets, illustrating better generalization ability than DAT. Specifically, MPL clearly outperforms DUST when there is only the acoustic mismatch, i.e., accent mismatch in CV dataset. And DUST has similar or better performance with MPL when there is also the linguistic mismatch, i.e., on TED and SWBD datasets. It illustrates that the online PL is more suitable for tackling the acoustic mismatch while the offline PL efficiently alleviates the linguistic mismatch by utilizing the target domain LM.

By utilizing advanced online and offline PL algorithms, both online and offline PL versions of CASTLE exhibit clear gains over their corresponding previous work. In particular, the online PL version of CASTLE surpasses the performance of robust wav2vec + MPL, and the offline version outperforms robust wav2vec + DUST. Notably, the online PL version already achieves better performance than existing approaches across all datasets, and the two-step version further enhances performance.

We also list some supervised results in the lower part of Table II to show the topline performance. As expected, the supervised results significantly outperform the unsupervised results. The continued pre-training is also shown to be effective in boosting supervised training performance. And using both source and target data for fine-tuning is slightly better than using only target data. CASTLE effectively closes the performance gap between the supervised and unsupervised approaches. Moreover, CASTLE could achieve comparable performance with supervised approaches on TED dataset, where the domain mismatch is the least.

We further give a direct comparison with results from related literature on the TED dataset. As shown in Table III, our approach significantly outperforms all previously reported unsupervised results and achieved competitive results with some supervised approaches.

TABLE III
DIRECT COMPARISON WITH PREVIOUS LITERATURE ON TED.

Method	External Data		WER%	
	Labeled	Unlabeled	Dev	Test
Unsupervised results				
DUST [24]	80h	×		17.6
MPL [25]	100h	×	16.2	14.9
Robust wav2vec [23]	10h	950h	8.9	-
CASTLE	100h	860h	**6.6**	**6.3**
Supervised results				
ESPnet-Conformer [83]	×	×	9.6	7.6
UniSpeech [84]	960h	×	7.6	7.6
SpeechStew [85]	4700h	×	-	5.3
SOTA [86]	9000h	×	**5.0**	**4.7**

In the following sections, we perform detailed experiments to analyze each component in CASTLE, where the most important ones contributing to the performance are: DPL, UCF and two-step PL.

B. Results of Continued Pre-training with Data Replay

In this section, we examine different continued pre-training strategies. Specifically, we compare (1) the source domain wav2vec 2.0 model, (2) continued pre-training the source model on target data, and (3) continued pre-training on target data while replaying source data. The continued pre-training updates for (2) and (3) are 20k. And the data replay ratio $p_{s/t}$ between source and target data is 0.5 for (3).

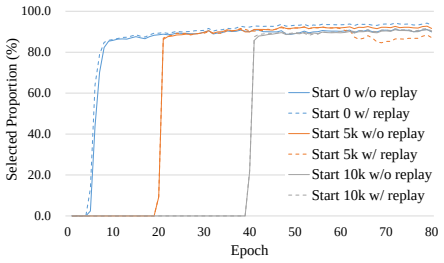
We evaluate pre-trained models with three fine-tuning strategies: (1) source-only: fine-tuning only on the labeled source data for 30k updates; (2) online PL: fine-tuning on both domains with the DPL approach for 20k updates; (3) offline PL: first fine-tuning on the labeled source data for 10k updates and then fine-tuning on the unlabeled target data with UCF-based offline PL for 4 iterations, where each iteration consists of 5k updates and target domain LM is always used when generating pseudo-labels.

TABLE IV
EFFECTIVENESS OF CONTINUED PRE-TRAINING STRATEGIES

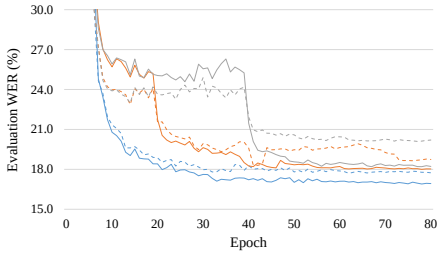
Pre-Train Method	WER% on RT03, Fine-Tune with		
	Source-Only	Offline PL	Online PL
Wav2vec 2.0	35.9	21.8	19.7
+ Continued pre-train	24.8	17.9	**16.9**
++ Data replay	**22.0**	**17.3**	17.7

As shown in Table IV, continued pre-training consistently improves the performance for three fine-tuning strategies. The

data replay technique further enhances the performance of the source-only fine-tuning by avoiding the catastrophic forgetting of source knowledge. And the offline PL performance has the same trend since source-only fine-tuning determines the quality of the seed model in offline PL. On the contrary, the online PL works better without data replay. We hypothesize that the reason is that simultaneously training on both domains could alleviate the catastrophic forgetting issue. Therefore, given a certain training update, continued pre-training on only the target domain leads to better target domain ability and better PL training on the target domain.



(a) Selected Proportion



(b) Evaluation WER

Fig. 6. Two-stage online PL behavior over epochs with different continued pre-training strategies and start updates. (a) Selected proportions of the unlabeled target domain training set. (b) Evaluation WER on the RT03 development set.

We verify the hypothesis with the two-stage DPL training, where the first stage is trained only with labeled source data and the PL loss is added in the second stage. As shown in Fig. 6, we conduct three sets of comparisons with different start update numbers (0, 5k, 10k) of the second stage. The model with data replay is consistently better in the first stage. In the second stage, the one without data replay quickly catches up and achieves better results. This phenomenon verifies that the model without data replay works better as long as the model is simultaneously fine-tuned on both domains.

In conclusion, different fine-tuning strategies require different continued pre-training strategies. When fine-tuning with source-only data, the SSL pre-trained model must incorporate knowledge of both domains to generalize to the target domain. Thus data replay could benefit source-only fine-tuning and offline PL. On the other hand, online PL does not suffer from the catastrophic forgetting of source knowledge and could benefit from concentrating on the target domain, i.e., without data replay. Therefore, CASTLE does not utilize data replay since the online PL approach DPL is used to fine-tune the SSL pre-trained model.

C. Results of Online PL with DPL

In this section, we conduct experiments to show the effectiveness of DPL. Firstly, we compare DPL with two other online PL approaches: vanilla online PL [59] and MPL [25]. All three approaches start from a continued pre-trained model without data replay and the fine-tuning updates are 20k. The hyper-parameters are tuned on the development set. Note that the two-stage training does not clearly improve the vanilla online PL. Thus we apply the single-stage training for it.

TABLE V
COMPARISON OF DIFFERENT ONLINE PL APPROACHES

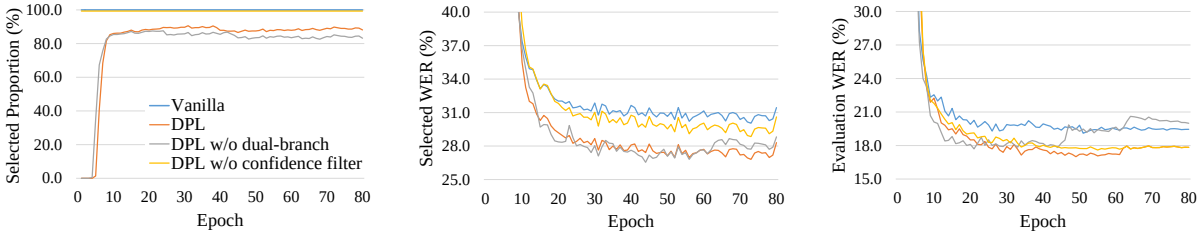
PL Method	WER%		
	RT03	H-SB	H-CH
Vanilla online PL	19.1	12.5	19.6
MPL	18.6	13.5	18.3
DPL	**16.9**	**11.7**	**17.6**
- Dual-branch	17.7	12.2	18.6
- Confidence filter	17.6	12.1	18.3

As shown in Table V, DPL significantly outperforms other two approaches. Compared with the vanilla online PL, DPL involves two essential components: dual-branch learning and confidence filtering. We conduct the ablation study for DPL and find the performance is worse if any component is removed. Furthermore, we illustrated the selected proportion and WER of pseudo-labels, as well as the evaluation WER on the development set in Fig. 7. Comparing DPL and DPL w/o confidence filter, the confidence filtering technique effectively reduces WER by filtering out the erroneous pseudo-labels. Comparing DPL and DPL w/o dual-branch, in the earlier updates, the one w/o dual-branch has a similar selected proportion and slightly better selected pseudo-labels, thus giving a better performance. Then, in the later updates, the one w/o dual-branch suffers from the error accumulation issue, leading to obvious performance degradation. In the meantime, the one w/ dual-branch gradually catches up and leads to better performance.

TABLE VI
VARIANTS OF DPL

PL Method	WER%		
	RT03	H-SB	H-CH
DPL	**16.9**	**11.7**	**17.6**
+ Pseudo-label from main branch	17.9	12.0	18.5
+ Two-stage Training	16.8	11.5	17.8
+ EMA (discount factor = 0.001)	19.4	13.7	19.8
+ Two-layer auxiliary branch	17.6	12.1	18.0

To further understand DPL, we show the results of some variants of DPL in Table VI. Firstly, we train a model that uses the auxiliary branch as a regularization. And the pseudo-labels are still from the main branch, like the vanilla online PL. The performance degradation of this model shows that generating pseudo-labels with the auxiliary branch is the key to the success of DPL. Secondly, instead of using confidence filtering to determine the utilized pseudo-labels automatically, we try the two-stage training where we manually choose when



(a) Selected Proportion (b) Selected WER (c) Evaluation WER

Fig. 7. Behaviors of different online PL approaches over epochs. (a) Selected proportions of the unlabeled target domain training set. (b) The WER of the selected training set. (c) Evaluation WER on the RT03 development set.

we start to use the PL loss. Like the vanilla online PL, DPL also does not benefit from the two-stage training. Thirdly, we add the EMA technique to DPL. We find a larger discount factor [60] (0.01 or 0.1), which gives more weight to the recent parameters, does not make a significant difference. And a smaller discount factor (0.001) slows down the convergence speed and leads to worse performance. Therefore, EMA is not necessary when the PL training is already stable. It is in line with the findings in [60], which illustrates that EMA could stabilize the PL training but lead to slow improvement. Lastly, we explore using more layers in the auxiliary branch. Specifically, we utilized one more transformer layer in addition to the last linear layer. The result shows that using only one linear layer is enough and performs better in practice.

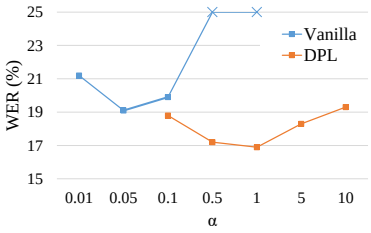


Fig. 8. Performance of different online PL approaches and different PL loss weights (α) on the RT03 development set. The cross symbol denotes the collapsed model trained to the trivial solution.

Next, we evaluate the DPL's sensitivity to the PL loss weight α in Eq. (4) . For comparison, we also evaluate the vanilla online PL. As shown in Fig. 8, when α is large, the vanilla online PL could lead to the trivial solution where the model predicts blank for any input. On the other hand, DPL is more stable and leads to consistently better performance.

Given the collapsed vanilla online PL training setup (α = 0.5) in Fig. 8, we illustrate how the confidence score in Eq. (6) could avoid the trivial solution. Since the proposed confidence score specifically discards the frames where the maximum score belongs to the blank, we also compare it with the variant that keeps blank in the computation.

As shown in Fig. 9, when no filtering strategy is applied, all pseudo-labels are used in training. The blank (or mostly blank) pseudo-labels can mislead the model to the trivial solution. Therefore, the selected WER and the evaluation WER quickly increase to 100% at a point.

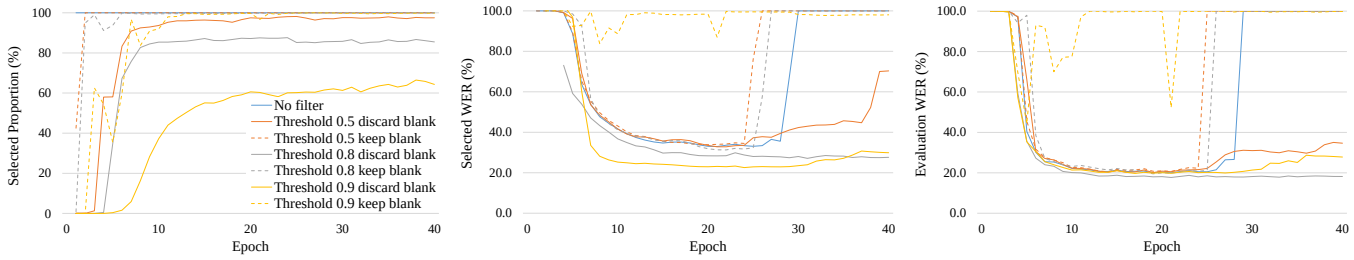
Then, we examine how confidence filtering affects the result. If we keep blanks when computing the confidence score, the blank (or mostly blank) pseudo-labels will have a high confidence score and will be kept in the selected subset. Consequently, the blank pseudo-labels would be more dominating in the selected subset, and the trivial solution happens even earlier than the one with no filtering. On the contrary, the confidence score that discards blank could benefit from more accurate selected pseudo-labels without the trouble of the blank pseudo-labels, thus being more stable and leading to better performance.

D. Results of Offline PL with Different Filtering Strategies

In this section, we conduct experiments to compare different filtering strategies for offline PL. Specifically, we consider confidence filtering [28], uncertainty filtering [24], and the proposed UCF. We first generate pseudo-labels with beam-search decoding and target domain LM. Then, we select some fixed proportions (10%, 30%, 50%, 70%, 100%) of the pseudo-labels with these filtering approaches and compute the WER of the selected subsets. Finally, we fine-tune the unadapted or the adapted model on the selected subsets for one iteration (5k updates) and evaluate the WER on the RT03 development set. Note that the unadapted model is the model fine-tuned only on source data on the basis of the source domain pre-trained model. And the adapted model is the model fine-tuned on both source and target data with online PL on the basis of the continued pre-trained model.

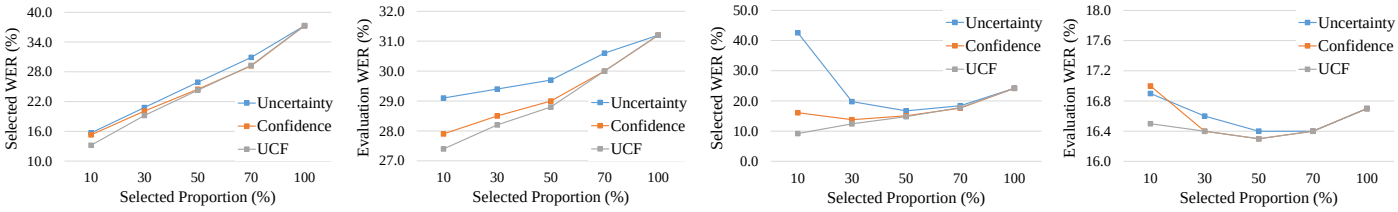
As shown in Fig. 10, for the unadapted model, since there are lots of errors in the pseudo-labels, no matter which filtering approach is used, the fewer pseudo-labels we select, the lower WER is obtained on the selected training subset. And the evaluation WER of the fine-tuned model is also decreased correspondingly. Among these filtering strategies, UCF is consistently better than others.

As for the adapted model (shown in Fig. 10), the confidence filtering and the uncertainty filtering could lead to a higher WER when we select a small proportion (10%) of pseudo-labels. And the evaluation WER increases correspondingly. This phenomenon indicates that the most certain or the most confident predictions are not the most accurate ones. On the contrary, UCF could effectively alleviate this issue by leveraging both criteria and estimating the hyper-parameters on the development set. Consequently, UCF significantly outperforms the other two filtering approaches when selecting a small proportion of pseudo-labels. The best evaluation WER is achieved when we select 50% of pseudo-labels with UCF or



(a) Selected Proportion (b) Selected WER (c) Evaluation WER

Fig. 9. Behaviors of vanilla online PL over epochs with different filtering strategies and confidence thresholds. (a) Selected proportions of the unlabeled target domain training set. (b) The WER of the selected training set. (c) Evaluation WER on the RT03 development set.



(a) Selected WER (unadapted) (b) Evaluation WER (unadapted) (c) Selected WER (adapted) (d) Evaluation WER (adapted)

Fig. 10. Behaviors of the unadapted and adapted models with different filtering strategies for offline PL. (a) and (c) are the WER of the selected training set. (b) and (d) are evaluation WER on the RT03 development set.

confidence filtering since the pseudo-labels from the adapted model are more accurate than the unadapted model.

E. Results of Two-Step PL

In this section, we evaluate the effectiveness of the two-step PL. For comparison, we list the performance of the one-step PL approaches, i.e., online or offline PL. Note that all approaches are trained with a total of 30k updates. To present the best results of the three approaches, the offline PL utilizes the data replay technique during continued pre-training, while the other two approaches do not.

TABLE VII
COMPARISON BETWEEN TWO-STEP PL AND ONE-STEP ONLINE/OFFLINE PL. SOURCE OR TARGET DOMAIN LM IS USED TO GENERATE PSEUDO-LABELS.

PL Method	LM Domain	WER%		
		RT03	H-SB	H-CH
Online PL	-	17.0	11.8	17.6
Offline PL	Target	17.3	11.2	17.6
Two-step PL	Source	16.4	11.2	17.4
	Target	**16.1**	**10.8**	**17.1**

As shown in Table VII, the two-step PL consistently outperforms both online and offline PL. Comparing two-step PL and online PL, the two-step PL improves the performance by generating more accurate pseudo-labels for refinement. Both source and target domain LM are beneficial and the target domain LM is better since it could transfer the target domain linguistic knowledge into the model. Note that the online PL does not benefit from more training update as the 30k updates gives similar results with the 20k updates in Table V.

TABLE VIII
GENERATE PSEUDO-LABELS WITH DECODING METHOD IN ONLINE PL W/O CONFIDENCE FILTERING.

Decode Method	LM weight	WER%		
		RT03	H-SB	H-CH
Greedy-decode	-	17.6	12.1	18.3
Beam-search	0	17.8	12.0	18.5
	0.5	17.9	12.1	18.9
	1.0	20.1	13.2	20.8

TABLE IX
GENERATE PSEUDO-LABELS WITH DECODING METHOD IN OFFLINE PL.

Decode Method	LM weight	WER%		
		RT03	H-SB	H-CH
Beam-search	0	19.8	13.8	19.8
	0.5	17.5	11.8	17.9
	1.0	17.3	11.2	17.6

Aside from two-step PL, there are two other ways to take advantage of LM: (1) using LM-based decoding to generate pseudo-labels during online PL; (2) utilizing offline PL from the seed model that is only supervised fine-tuned on source labeled data, i.e., pure offline PL w/o online PL.

In terms of the online PL, as shown in Table VIII, we empirically find that utilizing LM-based beam-search decoding to generate pseudo-labels in online PL can not lead to proper improvement. Previous studies [25], [69] have also opted for LM-free decoding in online PL with the concern that LM-based decoding might lead to over-fitting to LM. Therefore, we seek help from the offline PL to use LM, i.e., two-step PL. When LM is not used, greedy-decoding

provides comparable performance with beam-search but has less computation complexity, thus being a suitable choice for pseudo-label generation in online PL.

Regarding offline PL, using LM to produce better pseudo-labels is helpful (shown in Table IX). Nonetheless, employing online PL to generate a better seed model for offline PL, i.e., two-step PL, leads to better performance. Moreover, as shown in Fig. 11, merely increasing the number of iterations for offline PL cannot continue to reduce the WER after a certain iteration, and it cannot match the performance of two-step PL.

Therefore, we can safely conclude that online PL and offline PL are complementary and could be combined to achieve better performance.

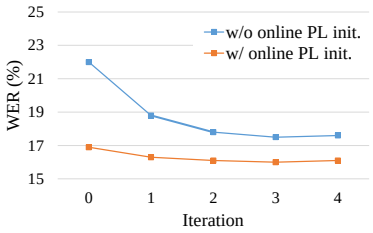


Fig. 11. Performance of offline PL over iterations on the RT03 development set. The seed model (iteration 0) is initialized w/ or w/o online PL.

VI. Discussions

In this section, we further discuss the proposed CASTLE approach and the possible future extensions.

- *Extending to other UDA settings:* We assume access to the target style text corpus and development set. But they might not be satisfied in some scenarios, thus requiring the modifications of the CASTLE approach. When the target style text corpus is unavailable, the source LM can be used instead in the offline PL. And when the development set is missing, UCF can degenerate to the confidence filtering approach, where γ and η are both 0 as they can't be automatically estimated on the development set.
- *Extending to other E2E-ASR models:* We demonstrated the effectiveness of the proposed approaches on CTC models in this work. But the proposed approaches are not model-specific and can be naturally extended to other E2E-ASR models. A future direction is to apply the idea of DPL, UCF and two-step PL on AED [87] and RNN-T [88] models with some possible modifications.
- *Extending to the model compression task:* We focused on improving the cross-domain performance and did not pay attention to model compression. Nonetheless, simple modifications of the CASTLE can extend to the model compression task: replace the student model with a smaller model in the last iteration of offline PL or conduct model compression to the final model of CASTLE.
- *Improving the confidence filtering for online PL:* Since the auxiliary forward computation of UCF would significantly increase the training time of online PL, UCF is only applied in the offline PL and we did not modify UCF to adapt the online PL scenario. However, the success of UCF on offline PL proves that a better confidence estimation can improve pseudo-labeling performance. Therefore, a future direction is to improve the confidence estimation in online PL without dramatically increasing the training time.
- *Integration with other complementary approaches:* CASTLE concentrates on the self-supervision based approaches to improve UDA performance. An intuitive complementary approach is target domain data synthesis, as more in-domain data is likely beneficial. However, the target data synthesis approach would complicate the training recipe and is less correlated with the proposed approach in this work. Therefore, we leave it as a future direction to be explored.

VII. Conclusion

In this work, we propose a systematic UDA approach CASTLE to utilize self-supervision for cross-domain speech recognition of the unlabeled target data. CASTLE is built based on SSL and PL in the pre-training and fine-tuning paradigm. It could effectively alleviate both pre-training and fine-tuning mismatch in the UDA scenario.

On the one hand, to deal with the pre-training mismatch, we dive into the continued pre-training and data replay techniques for better pre-training adaptation without the bother of catastrophic forgetting. We show that simultaneous fine-tuning on both domains could alleviate the trouble of catastrophic forgetting. And data replay would be significantly helpful if we only fine-tune the pre-trained model on labeled source data.

On the other hand, a domain adaptive fine-tuning approach is proposed to resolve a fine-tuning mismatch with PL, consisting of three unique techniques. Firstly, DPL alleviates the error accumulation of online PL; Secondly, UCF could select a better subset of pseudo-labels during offline PL; Lastly, the two-step PL improves the quality of pseudo-labels with LM.

Extensive ablation experiments are conducted to verify each component of the proposed approach. Experimental results demonstrate that CASTLE significantly outperforms previous UDA approaches and their naive combinations. Furthermore, CASTLE could achieve comparable performance with supervised training under minor domain mismatches.

Acknowledgment

This work is partially supported by the National Key Research and Development Program of China (No. 2020AAA0108002), the Youth Innovation Promotion Association, Chinese Academy of Sciences, and the Frontier Exploration Project Independently Deployed by Institute of Acoustics, Chinese Academy of Sciences under Grant QYTS202011.

References

[1] J. Li, "Recent advances in end-to-end automatic speech recognition," *APSIPA Transactions on Signal and Information Processing*, 2021.

[2] G. Cheng, H. Miao, R. Yang, K. Deng, and Y. Yan, "Eteh: Unified attention-based end-to-end asr and kws architecture," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 1360–1373, 2022.

[3] M. Huang, Y. Lu, L. Wang, Y. Qian, and K. Yu, "Exploring model units and training strategies for end-to-end speech recognition," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 524–531.

[4] H. Miao, G. Cheng, P. Zhang, and Y. Yan, "Online hybrid ctc/attention end-to-end automatic speech recognition architecture," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1452–1465, 2020.

[5] H. Miao, G. Cheng, C. Gao, P. Zhang, and Y. Yan, "Transformer-based online ctc/attention end-to-end speech recognition architecture," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6084–6088.

[6] P. Bell, J. Fainberg, O. Klejch, J. Li, S. Renals, and P. Swietojanski, "Adaptation algorithms for neural network-based speech recognition: An overview," *IEEE Open Journal of Signal Processing*, vol. 2, pp. 33–66, 2020.

[7] K. C. Sim, A. Narayanan, A. Misra, A. Tripathi, G. Pundak, T. N. Sainath, P. Haghani, B. Li, and M. Bacchiani, "Domain adaptation using factorized hidden layer for robust automatic speech recognition." in *Interspeech*, 2018, pp. 892–896.

[8] W. Hou, H. Zhu, Y. Wang, J. Wang, T. Qin, R. Xu, and T. Shinozaki, "Exploiting adapters for cross-lingual low-resource speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 317–329, 2021.

[9] J. Li, M. L. Seltzer, X. Wang, R. Zhao, and Y. Gong, "Large-scale domain adaptation via teacher-student learning," *Proc. Interspeech 2017*, pp. 2386–2390, 2017.

[10] W.-N. Hsu, Y. Zhang, and J. Glass, "Unsupervised domain adaptation for robust speech recognition via variational autoencoder-based data augmentation," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 16–23.

[11] E. Hosseini-Asl, Y. Zhou, C. Xiong, and R. Socher, "A multi-discriminator cyclegan for unsupervised non-parallel speech domain adaptation," *Proc. Interspeech 2018*, pp. 3758–3762, 2018.

[12] J. Li, R. Zhao, Z. Meng, Y. Liu, W. Wei, S. Parthasarathy, V. Mazalov, Z. Wang, L. He, S. Zhao *et al.*, "Developing rnn-t models surpassing high-performance hybrid models with customization capability," *Proc. Interspeech 2020*, pp. 3590–3594, 2020.

[13] M. K. Baskar, L. Burget, S. Watanabe, R. F. Astudillo *et al.*, "Eat: Enhanced asr-tts for self-supervised speech recognition," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6753–6757.

[14] F. Yue, Y. Deng, L. He, T. Ko, and Y. Zhang, "Exploring machine speech chain for domain adaptation," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6757–6761.

[15] W. Hou, J. Wang, X. Tan, T. Qin, and T. Shinozaki, "Cross-Domain Speech Recognition with Unsupervised Character-Level Distribution Matching," in *Proc. Interspeech 2021*, 2021, pp. 3425–3429.

[16] S. Sun, C.-F. Yeh, M.-Y. Hwang, M. Ostendorf, and L. Xie, "Domain adversarial training for accented speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4854–4858.

[17] B. Li, Y. Wang, T. Che, S. Zhang, S. Zhao, P. Xu, W. Zhou, Y. Bengio, and K. Keutzer, "Rethinking distributional matching based domain adaptation," *arXiv preprint arXiv:2006.13352*, 2020.

[18] H. Zhao, R. T. Des Combes, K. Zhang, and G. Gordon, "On learning invariant representations for domain adaptation," in *International Conference on Machine Learning*. PMLR, 2019, pp. 7523–7532.

[19] H. Liu, M. Long, J. Wang, and M. Jordan, "Transferable adversarial training: A general approach to adapting deep classifiers," in *International Conference on Machine Learning*. PMLR, 2019, pp. 4013–4022.

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "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 (Long and Short Papers)*, 2019, pp. 4171–4186.

[21] D. Hwang, A. Misra, Z. Huo, N. Siddhartha, S. Garg, D. Qiu, K. C. Sim, T. Strohman, F. Beaufays, and Y. He, "Large-scale asr domain adaptation using self-and semi-supervised learning," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6627–6631.

[22] A. Misra, D. Hwang, Z. Huo, S. Garg, N. Siddhartha, A. Narayanan, and K. C. Sim, "A comparison of supervised and unsupervised pre-training of end-to-end models," in *Proc. Interspeech*, vol. 2021, 2021, pp. 731–735.

[23] 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 *Proc. Interspeech 2021*, 2021, pp. 721–725.

[24] S. Khurana, N. Moritz, T. Hori, and J. Le Roux, "Unsupervised domain adaptation for speech recognition via uncertainty driven self-training," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6553–6557.

[25] Y. Higuchi, N. Moritz, J. L. Roux, and T. Hori, "Momentum Pseudo-Labeling for Semi-Supervised Speech Recognition," in *Proc. Interspeech 2021*, 2021, pp. 726–730.

[26] X. Jiang, Q. Lao, S. Matwin, and M. Havaei, "Implicit class-conditioned domain alignment for unsupervised domain adaptation," in *International Conference on Machine Learning*. PMLR, 2020, pp. 4816–4827.

[27] E. Arazo, D. Ortego, P. Albert, N. E. O'Connor, and K. McGuinness, "Pseudo-labeling and confirmation bias in deep semi-supervised learning," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–8.

[28] J. Kahn, A. Lee, and A. Hannun, "Self-training for end-to-end speech recognition," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7084–7088.

[29] D. S. Park, Y. Zhang, Y. Jia, W. Han, C.-C. Chiu, B. Li, Y. Wu, and Q. V. Le, "Improved noisy student training for automatic speech recognition," *Proc. Interspeech 2020*, pp. 2817–2821, 2020.

[30] Q. Li, D. Qiu, Y. Zhang, B. Li, Y. He, P. C. Woodland, L. Cao, and T. Strohman, "Confidence estimation for attention-based sequence-to-sequence models for speech recognition," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6388–6392.

[31] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*. PMLR, 2016, pp. 1050–1059.

[32] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith, "Don't stop pretraining: Adapt language models to domains and tasks," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 8342–8360.

[33] D. Hu, S. Yan, Q. Lu, L. HONG, H. Hu, Y. Zhang, Z. Li, X. Wang, and J. Feng, "How well does self-supervised pre-training perform with streaming data?" in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=EwqEx5ipbOu

[34] C. Anoop, A. Prathosh, and A. Ramakrishnan, "Unsupervised domain adaptation schemes for building asr in low-resource languages," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021, pp. 342–349.

[35] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by back-propagation," in *International conference on machine learning*. PMLR, 2015, pp. 1180–1189.

[36] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[37] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "Wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[38] S. Pascual, M. Ravanelli, J. Serrà, A. Bonafonte, and Y. Bengio, "Learning problem-agnostic speech representations from multiple self-supervised tasks," *Proc. Interspeech 2019*, pp. 161–165, 2019.

[39] Y.-A. Chung, W.-N. Hsu, H. Tang, and J. Glass, "An unsupervised autoregressive model for speech representation learning," *Proc. Interspeech 2019*, pp. 146–150, 2019.

[40] A. T. Liu, S.-W. Li, and H.-y. Lee, "Tera: Self-supervised learning of transformer encoder representation for speech," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2351–2366, 2021.

[41] W.-N. Hsu, Y.-H. H. Tsai, B. Bolte, R. Salakhutdinov, and A. Mohamed, "Hubert: How much can a bad teacher benefit asr pre-training?" in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6533–6537.

[42] S. wen Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin, T.-H. Huang, W.-C. Tseng, K. tik Lee, D.-R. Liu, Z. Huang, S. Dong, S.-W. Li, S. Watanabe, A. Mohamed, and H. yi Lee, "SUPERB: Speech Processing Universal PERformance Benchmark," in *Proc. Interspeech 2021*, 2021, pp. 1194–1198.

[43] Y.-A. Chung, Y. Zhang, W. Han, C.-C. Chiu, J. Qin, R. Pang, and Y. Wu, "w2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2021, pp. 244–250.

[44] C. Gao, G. Cheng, R. Yang, H. Zhu, P. Zhang, and Y. Yan, "Pre-training transformer decoder for end-to-end asr model with unpaired text data," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6543–6547.

[45] C. Gao, G. Cheng, T. Li, P. Zhang, and Y. Yan, "Self-supervised pre-training for attention-based encoder-decoder asr model," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2022.

[46] C. Gao, G. Cheng, Y. Guo, Q. Zhao, and P. Zhang, "Data augmentation based consistency contrastive pre-training for automatic speech recognition," *arXiv preprint arXiv:2112.12522*, 2021.

[47] J. Bai, B. Li, Y. Zhang, A. Bapna, N. Siddhartha, K. C. Sim, and T. N. Sainath, "Joint unsupervised and supervised training for multilingual asr," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6402–6406.

[48] C. Talnikar, T. Likhomanenko, R. Collobert, and G. Synnaeve, "Joint masked cpc and ctc training for asr," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 3045–3049.

[49] D. Hwang, K. C. Sim, Z. Huo, and T. Strohman, "Pseudo Label Is Better Than Human Label," in *Proc. Interspeech 2022*, 2022, pp. 1421–1425.

[50] A. Pasad, J.-C. Chou, and K. Livescu, "Layer-wise analysis of a self-supervised speech representation model," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021, pp. 914–921.

[51] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[52] B. Zhang, Y. Wang, W. Hou, H. Wu, J. Wang, M. Okumura, and T. Shinozaki, "Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[53] B. Chen, J. Jiang, X. Wang, P. Wan, J. Wang, and M. Long, "Debiased self-training for semi-supervised learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 32 424–32 437, 2022.

[54] F. Weninger, F. Mana, R. Gemello, J. Andrés-Ferrer, and P. Zhan, "Semi-supervised learning with data augmentation for end-to-end asr," *Proc. Interspeech 2020*, pp. 2802–2806, 2020.

[55] R. Masumura, M. Ihori, A. Takashima, T. Moriya, A. Ando, and Y. Shinohara, "Sequence-level consistency training for semi-supervised end-to-end automatic speech recognition," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7054–7058.

[56] B. Li, T. N. Sainath, R. Pang, and Z. Wu, "Semi-supervised training for end-to-end models via weak distillation," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 2837–2841.

[57] G. Synnaeve, Q. Xu, J. Kahn, T. Likhomanenko, E. Grave, V. Pratap, A. Sriram, V. Liptchinsky, and R. Collobert, "End-to-end asr: from supervised to semi-supervised learning with modern architectures," in *ICML workshop on Self-supervision in Audio and Speech (SAS)*, 2020.

[58] Q. Xu, T. Likhomanenko, J. Kahn, A. Hannun, G. Synnaeve, and R. Collobert, "Iterative pseudo-labeling for speech recognition," *Proc. Interspeech 2020*, pp. 1006–1010, 2020.

[59] Y. Chen, W. Wang, and C. Wang, "Semi-supervised asr by end-to-end self-training," *Proc. Interspeech 2020*, pp. 2787–2791, 2020.

[60] V. Manohar, T. Likhomanenko, Q. Xu, W.-N. Hsu, R. Collobert, Y. Saraf, G. Zweig, and A. Mohamed, "Kaizen: Continuously improving teacher using exponential moving average for semi-supervised speech recognition," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2021, pp. 518–525.

[61] H. Liao, E. McDermott, and A. Senior, "Large scale deep neural network acoustic modeling with semi-supervised training data for youtube video transcription," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE, 2013, pp. 368–373.

[62] P. Lanchantin, M. J. Gales, P. Karanasou, X. Liu, Y. Qian, L. Wang, P. C. Woodland, and C. Zhang, "The development of the cambridge university alignment systems for the multi-genre broadcast challenge," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2015, pp. 647–653.

[63] ——, "Selection of multi-genre broadcast data for the training of automatic speech recognition systems." in *Interspeech*, vol. 2016, 2016, pp. 3057–3061.

[64] Z. Zheng and Y. Yang, "Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation," *International Journal of Computer Vision*, vol. 129, no. 4, pp. 1106–1120, 2021.

[65] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," *Advances in neural information processing systems*, vol. 32, 2019.

[66] Z. Chen, A. Rosenberg, Y. Zhang, H. Zen, M. Ghodsi, Y. Huang, J. Emond, G. Wang, B. Ramabhadran, and P. J. Moreno, "Semi-Supervision in ASR: Sequential MixMatch and Factorized TTS-Based Augmentation," in *Proc. Interspeech 2021*, 2021, pp. 736–740.

[67] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[68] Y. Zhang, J. Qin, D. S. Park, W. Han, C.-C. Chiu, R. Pang, Q. V. Le, and Y. Wu, "Pushing the limits of semi-supervised learning for automatic speech recognition," *NeurIPS SAS 2020 Workshop*, 2020.

[69] T. Likhomanenko, Q. Xu, J. Kahn, G. Synnaeve, and R. Collobert, "slimIPL: Language-Model-Free Iterative Pseudo-Labeling," in *Proc. Interspeech 2021*, 2021, pp. 741–745.

[70] L. Lugosch, T. Likhomanenko, G. Synnaeve, and R. Collobert, "Pseudo-labeling for massively multilingual speech recognition," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7687–7691.

[71] Z. Chen, Y. Zhuang, and K. Yu, "Confidence measures for ctc-based phone synchronous decoding," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 4850–4854.

[72] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[73] F. Hernandez, V. Nguyen, S. Ghannay, N. Tomashenko, and Y. Esteve, "Ted-lium 3: twice as much data and corpus repartition for experiments on speaker adaptation," in *International conference on speech and computer*. Springer, 2018, pp. 198–208.

[74] J. Godfrey and E. Holliman, "Switchboard-1 release 2 ldc97s62," *Web Download. Philadelphia: Linguistic Data Consortium*, 1993.

[75] R. Ardila, M. Branson, K. Davis, M. Kohler, J. Meyer, M. Henretty, R. Morais, L. Saunders, F. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," in *Proceedings of the 12th Language Resources and Evaluation Conference*, 2020, pp. 4218–4222.

[76] e. a. Fiscus, Jonathan G., "2003 nist rich transcription evaluation data ldc2007s10," *Web Download. Philadelphia: Linguistic Data Consortium*, 2007.

[77] L. D. Consortium, "2000 hub5 english evaluation speech ldc2002s09," *Web Download. Philadelphia: Linguistic Data Consortium*, 2002.

[78] A. Rousseau, P. Deléglise, Y. Esteve *et al.*, "Enhancing the ted-lium corpus with selected data for language modeling and more ted talks." in *LREC*, 2014, pp. 3935–3939.

[79] e. a. Cieri, Christopher, "Fisher english training part 2, transcripts ldc2005t19," *Web Download. Philadelphia: Linguistic Data Consortium*, 2005.

[80] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, "fairseq: A fast, extensible toolkit for sequence modeling," in *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

[81] 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," *Proc. Interspeech 2019*, pp. 2613–2617, 2019.

[82] D. Berthelot, R. Roelofs, K. Sohn, N. Carlini, and A. Kurakin, "Adamatch: A unified approach to semi-supervised learning and domain adaptation," in *International Conference on Learning Representations*, 2021.

[83] P. Guo, F. Boyer, X. Chang, T. Hayashi, Y. Higuchi, H. Inaguma, N. Kamo, C. Li, D. Garcia-Romero, J. Shi *et al.*, "Recent developments on espnet toolkit boosted by conformer," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5874–5878.

[84] C. Wang, Y. Wu, Y. Qian, K. Kumatani, S. Liu, F. Wei, M. Zeng, and X. Huang, "Unispeech: Unified speech representation learning with labeled and unlabeled data," in *International Conference on Machine Learning*. PMLR, 2021, pp. 10 937–10 947.

[85] W. Chan, D. Park, C. Lee, Y. Zhang, Q. Le, and M. Norouzi, "Speechstew: Simply mix all available speech recognition data to train one large neural network," *arXiv preprint arXiv:2104.02133*, 2021.
[86] T. Likhomanenko, Q. Xu, V. Pratap, P. Tomasello, J. Kahn, G. Avidov, R. Collobert, and G. Synnaeve, "Rethinking Evaluation in ASR: Are Our Models Robust Enough?" in *Proc. Interspeech 2021*, 2021, pp. 311–315.
[87] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.
[88] A. Graves, "Sequence transduction with recurrent neural networks," in *ICML Workshop on Representation Learning*, 2012.