Adversarial Meta Sampling for Multilingual Low-Resource Speech Recognition

Yubei Xiao[1], Ke Gong[3], Pan Zhou[4], Guolin Zheng[1], Xiaodan Liang[2,3], Liang Lin[1,3*]

[1] School of Computer Science and Engineering, Sun Yat-sen University, China
[2] School of Intelligent Systems Engineering, Sun Yat-sen University, China
[3] Dark Matter AI Research, [4] SalesForce
{xiaoyb5,zhengglin}@mail2.sysu.edu.cn, {kegong936,xdliang328}@gmail.com, pzhou@salesforce.com, linliang@ieee.org

Abstract

Low-resource automatic speech recognition (ASR) is challenging, as the low-resource target language data cannot well train an ASR model. To solve this issue, meta-learning formulates ASR for each source language into many small ASR tasks and meta-learns a model initialization on all tasks from different source languages to access fast adaptation on unseen target languages. However, for different source languages, the quantity and difficulty vary greatly because of their different data scales and diverse phonological systems, which leads to task-quantity and task-difficulty imbalance issues and thus a failure of multilingual meta-learning ASR (MML-ASR). In this work, we solve this problem by developing a novel adversarial meta sampling (AMS) approach to improve MML-ASR. When sampling tasks in MML-ASR, AMS adaptively determines the task sampling probability for each source language. Specifically, for each source language, if the query loss is large, it means that its tasks are not well sampled to train ASR model in terms of its quantity and difficulty and thus should be sampled more frequently for extra learning. Inspired by this fact, we feed the historical task query loss of all source language domain into a network to learn a task sampling policy for adversarially increasing the current query loss of MML-ASR. Thus, the learnt task sampling policy can master the learning situation of each language and thus predicts good task sampling probability for each language for more effective learning. Finally, experiment results on two multilingual datasets show significant performance improvement when applying our AMS on MML-ASR, and also demonstrate the applicability of AMS to other low-resource speech tasks and transfer learning ASR approaches.

Introduction

Automatic Speech Recognition (ASR) has attracted a lot of attention recently and achieved significant improvements (Chan et al. 2016; Graves et al. 2006; Pratap et al. 2019) brought by the success of deep neural networks. However, building an end-to-end deep ASR model often requires huge transcribed training data, which is impractical for the low-resource languages due to the scarcity of audio data and the huge labor resources consumed in transcription.

To solve this issue, many works are devoted to develop low-resource ASR approaches. The representative methods in this line are transfer learning ASR (**TL-ASR**) (Hu et al. 2019; Kunze et al. 2017), multilingual transfer learning ASR (**MTL-ASR**) (Adams et al. 2019; Cho et al. 2018; Tong, Garner, and Bourlard 2017) and multilingual meta-learning ASR (**MML-ASR**) (Hsu, Chen, and yi Lee 2020) that all aim to learn an ASR model initialization from source languages such that the initialization can quickly adapt to target language via fine-tuning on a few data. Among them, TL-ASR considers one source language and regards the pretrained ASR model on the source data as a model initialization. But as shown in Fig. 1 (a), the learnt initialization by TL-ASR often overfits the source language and cannot quickly adapt to a different target language. To resolve this issue, MTL-ASR and MML-ASR consider multiple source languages. Inspired by multi-task learning, they both sample partial data from each language domain to construct a small speech recognition **task**. Then for each sampled task, MTL-ASR directly trains its model on this task, while MML-ASR adapts its ASR model to the validation data of the task via fine-tuning on a few training data of the task and minimizes the validation loss. In this way, the learnt initializations by MTL-ASR and MML-ASR can usually fast adapt to the target low-resource language, as both MTL-ASR and MML-ASR learn the common knowledge from all tasks from different language domains which facilitates learning target languages.

However, when sampling tasks from these language domains, MTL-ASR and MML-ASR often ignore the underlying task imbalance issues which could result in unsatisfactory performance. First, different kinds of languages have different training data scales so the underlying task quantity for each language domain varies greatly, which leads to the **task-quantity imbalance**. Second, as different languages have diverse phonological systems, the tasks drawn from different language domains have various recognition difficulties, causing the **task-difficulty imbalance**. In this way, for MTL-ASR and MML-ASR, both **uniform sampling** that uniformly samples tasks from each language domain and more advanced **task-quantity-balanced sampling** whose sampling rate for each domain positively relies on its task quantity (data scale) cannot handle the task imbalance issue. Uniform sampling neither considers the task-quantity imbalance nor the task-difficulty imbalance, while task-quantity-balanced sampling directly ignores task-difficulty imbalance. So the learnt initializations by MTL-ASR and MML-ASR

*Liang Lin is the corresponding author of this work.

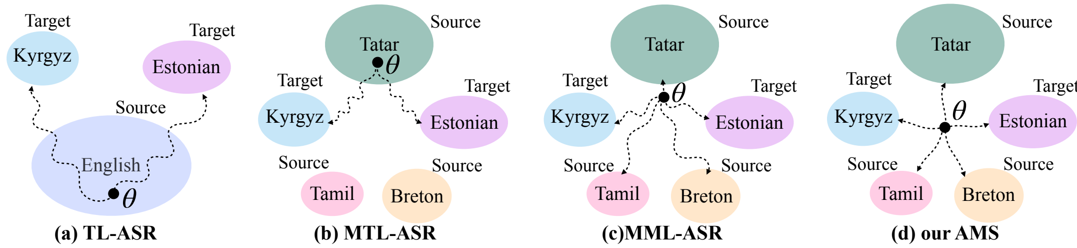


Figure 1: Comparison of learnt initializations under task-quantity-balanced sampling. The dashed lines denote the adaptation paths from initialization θ to target languages and the circular area of the language represents the training data scale of the language. (a) Initialization learnt in TL-ASR overfits the only one source language. Initializations in MTL-ASR (b) and MML-ASR (c) are close to the optimal model of the large-scaled language and departure from the small-scaled languages. (d) Initialization learnt by our AMS has a more balanced distance to all languages because of our adaptive sampling to handle the task imbalance.

are biased, and cannot fast adapt to the target language. For instance, as shown in Fig. 1 (b) and (c), when using task-quantity-balanced sampling in MTL-ASR and MML-ASR, their learnt initializations are often close to the optimal model of the large-scaled language and departure from those of the small-scaled languages. So the learnt initializations are often far from the optimal models of target languages which usually locates around all source languages, and cannot be fast adapted to target languages via fine-tuning on low-resource training data.

Contributions. To resolve the above task imbalance issue, we develop a novel adversarial meta sampling (**AMS**) method for multilingual low-resource speech recognition. This AMS can effectively help both MML-ASR and MLT-ASR handle task imbalance problem and boost their performance. Considering the superior performance of MML-ASR over MLT-ASR, in this work we spend more efforts to introduce AMS on MML-ASR. Specifically, we observe that the query losses of tasks from each language domain can well measure both task-quantity imbalance and task-difficulty imbalance. For each language domain, if its tasks are not well sampled in terms of its task quantity and task difficulty, then its tasks will have relatively large query losses. It means that the language domain with large task query loss requires more training. So we design a policy network to increase the query loss of MML-ASR model through adversarial learning for sampling from proper language domain. Our policy network incorporates LSTM (Hochreiter and Schmidhuber 1997) structure and attention mechanism to adaptively predict the most befitting task sampling probability for each language domain by using the long-term information in LSTM and the current query losses at each training iteration. Through such an online and adversarial manner, the sampling policy is dynamically changed along with the training state of the MML-ASR. In this way, the language domain that are not well sampled in terms of its task quantity and task difficulty will be sampled more in the later training iterations, making the MML-ASR model learn a more balanced initialization for better adaptation to the target languages as shown in Fig. 1 (d).

Moreover, we validate our method on several datasets with diverse languages selected from Mozilla Common Voice Corpus (Ardila et al. 2020) and the public IARPA BABEL dataset (Gales et al. 2014). The experimental results demonstrate that our AMS significantly improves the performance over the existing approaches on low-resource ASR, especially under the realistic task-imbalance scenarios. Furthermore, we conduct experiments on speech classification and speech translation, which proves that our AMS can be easily generalized to improve other low-resource speech tasks.

Related Work

Transfer learning ASR. To alleviate the need for labeled data, recent works utilize unsupervised pre-training and semi-supervised methods to exploit unlabeled data, *e.g.* wav2vec (Schneider et al. 2019), predictive coding (Chung and Glass 2020), self-training (Kahn, Lee, and Hannun 2020) and weak distillation (Li et al. 2019). But they still require substantial unlabeled data which is unavailable for some minority languages. To solve this issue, transfer learning is explored via using other source languages to improve the performance of low-resource languages (Kunze et al. 2017), which requires that the source and target languages are similar and the source language has sufficiently large data. Moreover, multilingual transfer learning ASR (Dalmia et al. 2018; Watanabe, Hori, and Hershey 2017; Toshniwal et al. 2018) is developed using different languages to learn language-independent representations for performance improvement under the low-resource setting.

Meta-learning ASR. Meta-learning approaches (Zhou et al. 2019, 2020) can meta-learn a model initialization from training tasks with fast adaptation ability to new tasks with only a few data and thus is suitable to handle low-resource data learning problems. Especially, Hsu *et al.*(Hsu et al. 2020) and Winata *et al.* (Winata et al. 2020) adopted MAML (Finn et al. 2017) for low-resource ASR and code-switched ASR and both achieved promising results. But these method ignores task imbalance in real-world scenarios and equally utilizes the meta-knowledge across all the languages, which leads to performance degradation. To alleviate quantity imbalance, Wang *et al.* (Wang, Tsvetkov, and Neubig 2020) improves differentiable data selection by optimizing a scorer with the average loss from different languages to balance the usage of data in multilingual model training. Besides the language quantity, our AMS also considers the language difficulty and learns the sampling policy in an adversarial manner.

Adversarial learning ASR. Inspired by domain adversarial training (Ganin et al. 2016), recent works introduced adversarial learning into ASR to learn robust features invariant to noise conditions (Shinohara 2016) and accents (Sun et al. 2018b). Besides, some researchers use a domain-adversarial classification objective over many languages on multilingual ASR framework to force the shared layers to learn language-independent representations (Yi et al. 2018). Differently, our proposed method explores adversarial learning to solve the task imbalance problem in multilingual meta-learning ASR and can learn to adaptively sample the meta-training tasks for effectively training low-resource ASR models.

Preliminaries

Here we briefly introduce the ASR model and its meta-learning version which are used in our method.

Multilingual Speech Recognition

ASR model. We first introduce the joint CTC-attention based end-to-end ASR architecture (Kim, Hori, and Watanabe 2017; Hori et al. 2017) because of its effectiveness and efficiency. It consists of a Seq2Seq network for frames alignment and symbols recognition, and a connectionist temporal classification (CTC) module (Graves et al. 2006) to encourage the alignments to be monotonic. For the seq2seq model, it contains an encoder, a decoder, and an attention unit. CTC is on the top of the encoder and is jointly trained with the Seq2Seq model. Then ASR network combines these two components and minimizes $\mathcal{L}(\theta) = \lambda_{\text{ctc}}\mathcal{L}_{\text{ctc}} + (1 - \lambda_{\text{ctc}})\,\mathcal{L}_{\text{seq2seq}}$.

Multilingual ASR model. To overcome the challenges brought by different sub-word units, lexicon and word inventories between different languages, we take the union over all the language-specific token sets and train a single model on a mixture dataset which combines all the source language data (Watanabe, Hori, and Hershey 2017; Toshniwal et al. 2018). Given N languages with training sets $\{D_i\}_{i=1}^N$ and token sets $\{C_i\}_{i=1}^N$, the mixture training set is $\mathcal{D}_{\text{multilingual}} = \cup_{i=1}^N D_i$ and the token set for the mixture dataset is $\mathcal{C}_{\text{multilingual}} = \cup_{i=1}^N C_i$.

Multilingual Meta-learning ASR

We train a multilingual meta-learning ASR (**MML-ASR**) model on all languages to pursue the few-shot learning ability to handle the low resource recognition problems. Specially, we use $f(\theta)$ to denote a multilingual ASR model parameterized by θ and adopt $\mathcal{D}_{\text{source}} = \{D^k_{\text{source}}\}_{k=1}^K$ to denote K kinds of source languages. To apply meta learning, *e.g.* MAML (Finn, Abbeel, and Levine 2017) and Reptile (Nichol, Achiam, and Schulman 2018), for the k-th kind of language, we sample partial data D^k_{task} (a few sentences) from D^k_{source} to construct a small recognition **task** $\mathcal{T}^i_k$. Then we split D^k_{task} into **support data** D^k_{support} and **query data** D^k_{query}. Accordingly, for each language we can sample many tasks denoted by $\mathcal{T}_k = \{\mathcal{T}^i_k\}_{i=1}^{n_k}$, where n_k is the **task quantity** of the k-th kind of language. Let V be the total number of examples in the k-th kind of language and w be the number of examples per task, then n_k could be calculated by combination number C^w_V. For brevity, let $\mathcal{T}$ be the all task set $\mathcal{T} = \{\{\mathcal{T}^i_1\}_{i=1}^{n_1}, \cdots, \{\mathcal{T}^i_K\}_{i=1}^{n_K}\}$. Now MML-ASR can be formulated as:

$$\min_\theta \; \mathbb{E}_{\mathcal{T}_i \sim \mathcal{T}} \mathcal{L}_{D_{\text{query}}}(\theta - \alpha\nabla_\theta \mathcal{L}_{D_{\text{support}}}(\theta)), \qquad (1)$$

where $\mathcal{T}_i$ is sampled from $\mathcal{T}$, D_{support}, and D_{query} respectively denote the support and query data in task $\mathcal{T}_i$. This model can be understood as that given a common model parameter θ for all language (or tasks), for a sampled task $\mathcal{T}_i$, we adapt the parameter θ to this specific task via running one gradient descent on its support data and obtain the task specific model parameter $\theta_{\mathcal{T}_i} = \theta - \alpha\nabla_\theta\mathcal{L}_{D_{\text{support}}}(\theta)$. Then we evaluate the effectiveness of $\theta_{\mathcal{T}_i}$ on the query data D_{query} of task $\mathcal{T}_i$ and use this **query loss**, denoted as $Q_{\mathcal{T}_i} = \mathcal{L}_{D_{\text{query}}}(\theta_{\mathcal{T}_i})$, to guide the learning of the model parameter θ. This process actually requires the learnt common model parameter θ to be close to the optimal model of all tasks $\mathcal{T}_i$ in task set $\mathcal{T}$ such that taking only one gradient step on a small-sized dataset D_{support} can achieve satisfactory performance on the query data D_{query}. This mechanism gives the few shot learning ability of model parameter θ.

After training, given a new speech recognition task with a few training data, we adapt the model parameter θ to this task by a few gradient descent steps and obtain a task specific model for test. So this MML-ASR model can well handle the low resource speech recognition problem.

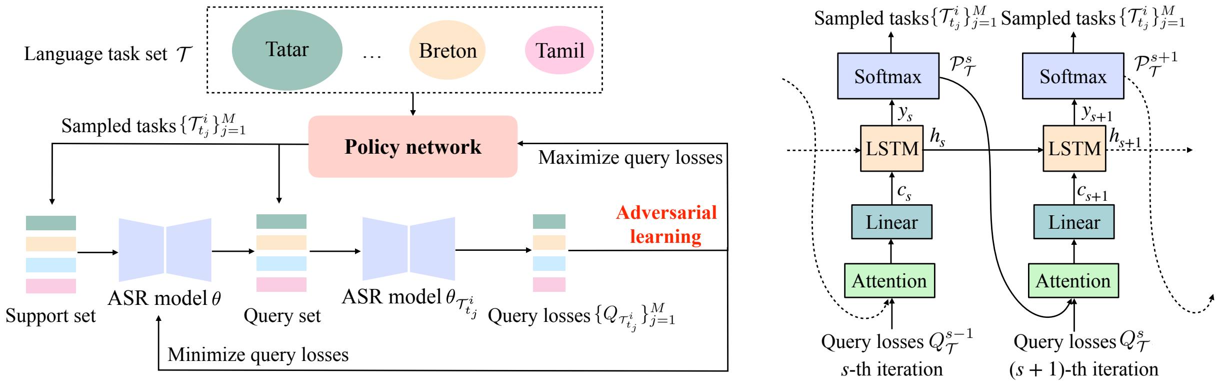
This MML-ASR is inspired by the prior meta-learning ASR framework (Hsu, Chen, and yi Lee 2020). Hsu *et al.* (Hsu, Chen, and yi Lee 2020) first used a shared backbone to extract common features for all languages and then adopted different network branches to learn language-specific features, while MML-ASR here uses an entire shared model for all languages to simplify operations and make full use of information from different languages.

Adversarial Meta Sampling

Motivation

As introduced in Sec. , for each meta-training iteration, we need to sample a task $\mathcal{T}_i$ from the task set $\mathcal{T} = \{\{\mathcal{T}^i_1\}_{i=1}^{n_1}, \cdots, \{\mathcal{T}^i_K\}_{i=1}^{n_K}\}$ where $\mathcal{T}_k = \{\mathcal{T}^i_k\}_{i=1}^{n_k}$ denotes the tasks in the k-th language domain. In real-world scenarios, different languages have a diverse geographic location, phonology, phonetic inventory, language family, and orthography, and their datasets vary greatly in size. So when sampling task $\mathcal{T}_i$ from all languages task set $\mathcal{T}$ to train our MML-ASR model, there are two severe issues. The first one is that the underlying task quantity $(n_1, \cdots, n_K)$ of each language fluctuates over a wide range, which means that the task sets $\mathcal{T}_k$ $(k = 1, \cdots, K)$ are imbalanced in terms of their task quantity (**task-quantity imbalance**). The second issue is that the tasks sampled from different language task sets $\mathcal{T}_k$ $(k = 1, \cdots, K)$ actually have very different recognition difficulty due to the aforementioned language specificities (Waibel et al. 2000), leading to an imbalance in terms of task difficulty (**task-difficulty imbalance**). So sampling approaches become especially important in ASR. Note, uniform sampling or task-quantity-balanced sampling, namely sampling rate for each language positively relying on its task quantity in $\mathcal{T}_k$, usually ignore the task-difficulty imbalance, and cannot achieve satisfactory performance as shown in our experiments.

To resolve above two imbalance issues, we propose a novel and effective adversarial meta sampling approach that adap-



(a) Our AMS framework

(b) Policy network

Figure 2: (a) The illustration of our AMS framework. (b) The architecture of the policy network.

tively determines the sampling probability for each language task set $\mathcal{T}_k$ in the meta-training process to balance both task quantity and difficulty in different language domains. Specifically, we observe that the query losses of tasks can well measure both imbalances because that if one language task set $\mathcal{T}_k$ are not well sampled in terms of its task quantity and task difficulty, then its tasks have relatively large query loss. Intuitively, at each iteration, one can sample several tasks from each language tasks $\mathcal{T}_k$ and compute the average query loss $Q_{\mathcal{T}_k}$ for each $\mathcal{T}_k$. Then a simple way is to use $Q_{\mathcal{T}_k} / \sum_{i=1}^{K} Q_{\mathcal{T}_k}$ as the sampling probability for each $\mathcal{T}_k$. However, this method ignores the long-term query loss information and only greedily assigns a large probability to some language tasks $\mathcal{T}_k$ according to the current query loss, which could be too locally greedy and leads to performance degradation. Maintaining a query loss buffer which linearly or exponentially averages the historical query losses still cannot achieve satisfactory performance, since the importance of current query loss and historical query losses is not necessarily a simple (exponential) average relation. Moreover, it requires to tune several manual hyper-parameters (*e.g.*, window size) for computing average query losses. All sampling methods mentioned above have been evaluated in Table 4.

Our Sampling Approach

To solve the above issues, we propose adversarial meta sampling method by designing a policy network which injects attention mechanism into LSTM (Hochreiter and Schmidhuber 1997). At each training iteration, it can adaptively predict the most befitting probability to sample from each language tasks $\mathcal{T}_k$ by using the long-term information in LSTM and the current query losses. Moreover, the policy network can be jointly trained with MML-ASR model in an end-to-end way without manual tuning extra hyper-parameters.

Specifically, as shown in Fig. 2 (a), at each meta-training iteration, the policy network samples M kinds of language task set denoted by $\{\mathcal{T}_{t_j}\}_{j=1}^M, t_j \in \{1, \cdots, K\}$ from the K kinds of language task $\mathcal{T}$ and then samples one task $\mathcal{T}_{t_j}^i$ from each $\mathcal{T}_{t_j}$ to form training task set $\{\mathcal{T}_{t_j}^i\}_{j=1}^M$ for meta-training of MML-ASR model. So the meta-objective of MML-ASR model in Eqn. (1) can be reformulated as

$$\min_\theta \; \mathbb{E}_{\pi \sim f_\phi} \mathbb{E}_{\mathcal{T}_i \sim \pi(\mathcal{T})} \mathcal{L}_{D_{\text{query}}}(\theta - \alpha \nabla_\theta \mathcal{L}_{D_{\text{support}}}(\theta)), \quad (2)$$

where π denotes the task sampling policy learnt by the policy network f_ϕ parameterized by ϕ.

After meta-training of MML-ASR model, we can obtain the query losses $\{Q_{\mathcal{T}_{t_j}^i}\}_{j=1}^M$ of each training task $\{\mathcal{T}_{t_j}^i\}_{j=1}^M$. As mentioned in Sec. , the query losses of tasks drawn from each language task set $\mathcal{T}_k$ can well measure the task imbalances in terms of both task quantity and task difficulty. This is because if one language task set $\mathcal{T}_k$ are not well sampled in terms of its task quantity and task difficulty, then its tasks have relatively large query loss. This actually means that the language task set $\mathcal{T}_k$ with large query loss $Q_{\mathcal{T}_k}$ requires more extra training. So at each iteration, our policy network attempts to increase the query loss of MML-ASR model through adversarial learning for sampling the proper language task set for training. Formally, the objective loss of our policy network is defined as:

$$\begin{aligned} &\phi^* = \arg\max_\phi \mathcal{J}(\phi), \text{where} \mathcal{J}(\phi) = \\ &\mathbb{E}_{\pi \sim f_\phi} \mathbb{E}_{\mathcal{T}_i \sim \pi(\mathcal{T})} \mathcal{L}_{D_{\text{query}}}(\theta - \alpha \nabla_\theta \mathcal{L}_{D_{\text{support}}}(\theta)). \end{aligned} \quad (3)$$

Then we focus on introducing the policy network f_ϕ, which shown in Fig. 2 (b). First, we use K-dim vector $Q_{\mathcal{T}}^{s-1} = (Q_{\mathcal{T}_1}^{s-1}, Q_{\mathcal{T}_2}^{s-1}, ..., Q_{\mathcal{T}_K}^{s-1})$ to denote the query loss for each language task set $\mathcal{T}_k$ at the $(s-1)$-th meta-training iteration. Second, at the s-th iteration, the policy network will output a K-dim $\mathcal{P}_{\mathcal{T}}^s = (\mathcal{P}_{\mathcal{T}_1}^s, \mathcal{P}_{\mathcal{T}_2}^s, ..., \mathcal{P}_{\mathcal{T}_K}^s)$ in which $\mathcal{P}_{\mathcal{T}_k}^s$ denotes the sampling probability for the k-th language task set $\mathcal{T}_k$. Third, as aforementioned, we select the top-M sampling probabilities and respectively sample one task from their corresponding language task set $\mathcal{T}_k$ for meta-training. Then we use the M new query loss $\{Q_{\mathcal{T}_{t_j}^i}^s\}_{j=1}^M$ to update the corresponding query loss in $Q_{\mathcal{T}}^{s-1}$ for obtaining $Q_{\mathcal{T}}^s$.

Next, at the $(s+1)$-th iteration, we feed $Q_{\mathcal{T}}^s$ and $\mathcal{P}_{\mathcal{T}}^s$ into the policy network and combine these two inputs $(Q_{\mathcal{T}}^s, \mathcal{P}_{\mathcal{T}}^s)$ to calculate the feed forward attention, and then get the attention output c_{s+1} through a fully-connected layer. Then, a LSTM layer takes the hidden state of previous LSTM cell h_s as well as attention output c_{s+1} as input, and generates the LSTM output y_{s+1} and current hidden state h_{s+1}. Finally, based on

Algorithm 1 Adversarial Meta Sampling

Require: α, β, γ: step size hyperparameters
Initialize θ, ϕ
Initialize $Q_{\mathcal{T}_k} = 0, \forall k \in \{1, 2, ..., K\}$
while not done **do**
  Generate K-dim vector of sampling probabilities $\mathcal{P}_{\mathcal{T}} = (\mathcal{P}_{\mathcal{T}_1}, \mathcal{P}_{\mathcal{T}_2}, ..., \mathcal{P}_{\mathcal{T}_K})$ using f_ϕ
  Sample M language task set $\{\mathcal{T}_{t_j}\}_{j=1}^M, t_j \in \{1, \cdots, K\}$ with top-M largest sampling probabilities
  Sample one task $\mathcal{T}_{t_j}^i$ from each $\mathcal{T}_{t_j}$ to form $\{\mathcal{T}_{t_j}^i\}_{j=1}^M, t_j \in \{1, \cdots, K\}$ for meta-training
  for all $\mathcal{T}_{t_j}^i$ **do**
    Generate support set $D_{\text{support}}^{t_j}$ and query set $D_{\text{query}}^{t_j}$ from $\mathcal{T}_{t_j}^i$
    Compute adapted parameters with respect to $D_{\text{support}}^{t_j}$ using $\theta_{\mathcal{T}_{t_j}^i} = \theta - \alpha \nabla_\theta \mathcal{L}_{D_{\text{support}}^{t_j}}(\theta)$
  end for
  Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{j=1}^M Q_{\mathcal{T}_{t_j}^i}$, where $Q_{\mathcal{T}_{t_j}^i} = \mathcal{L}_{D_{\text{query}}^{t_j}}(\theta_{\mathcal{T}_{t_j}^i})$ using each $D_{\text{query}}^{t_j}$
  Update query loss $Q_{\mathcal{T}} = (Q_{\mathcal{T}_1}, ..., Q_{\mathcal{T}_K})$, where $Q_{\mathcal{T}_k} = Q_{\mathcal{T}_{t_j}^i}|_{t_j=k}$ if $\mathcal{T}_k$ is sampled else $Q_{\mathcal{T}_k} = Q_{\mathcal{T}_k}^{s-1}$
  Update $\phi \leftarrow \phi + \gamma \nabla_\phi \sum_{j=1}^M \mathcal{P}_{\mathcal{T}_{t_j}} \mathcal{L}_{D_{\text{query}}^{t_j}}(\theta - \alpha \nabla_\theta \mathcal{L}_{D_{\text{support}}^{t_j}}(\theta))$
end while

y_{s+1}, we use fully-connected layer with Softmax function to predict a probability vector $\mathcal{P}_{\mathcal{T}}^{s+1} = (\mathcal{P}_{\mathcal{T}_1}^{s+1}, \mathcal{P}_{\mathcal{T}_2}^{s+1}, ..., \mathcal{P}_{\mathcal{T}_K}^{s+1})$. In this way, same as the s-th iteration, we can select the top-M largest probabilities and sample tasks from their corresponding task sets for meta-training.

As the discrete sampling operations for obtaining M tasks is not differentiable, we apply REINFORCE algorithm (Williams 1992) to solve this issue and optimize the policy network via the following gradient,

$$\nabla_\phi \mathcal{J}(\phi) = \nabla_\phi \mathbb{E}_{\pi \sim f_\phi} \mathbb{E}_{\mathcal{T}_i \sim \pi(\mathcal{T})} \mathcal{L}_{D_{\text{query}}}(\theta - \alpha \nabla_\theta \mathcal{L}_{D_{\text{support}}}(\theta))$$
$$\approx \nabla_\phi \sum_{i=1}^M \mathcal{P}_{\mathcal{T}_i} \mathcal{L}_{D_{\text{query}}}(\theta - \alpha \nabla_\theta \mathcal{L}_{D_{\text{support}}}(\theta)).$$

Through such an online and adversarial manner, the sampling policy is dynamically changed along with the training state of the MML-ASR. In this way, the language task set $\mathcal{T}_k$ that is not well sampled in terms of its task quantity and task difficulty will be sampled more for more effective learning.

Moreover, *our sampling method can be applied to not only MML-ASR methods but also multilingual transfer learning ASR (**MTL-ASR**) without any architecture modification*. For MTL-ASR, we directly use the training loss $\mathcal{L}_{\mathcal{T}_k}$ of a task sampled from each source language domain to construct the loss vector $(\mathcal{L}_{\mathcal{T}_1}, \mathcal{L}_{\mathcal{T}_2}, ..., \mathcal{L}_{\mathcal{T}_K})$ and feed it into our policy network. Similarly, the policy network outputs the sampling probability for each language domain. Experimental results in Sec. also verify the superiority of our policy network and show significant improvement when applying our sampling method into MTL-ASR.

Finally, our Adversarial Meta Sampling framework is summarised in Algorithm 1.

Experiments

Datasets. Common Voice (Ardila et al. 2020) is an open-source multilingual voice dataset and contains about 40 kinds of languages. For low-resource evaluation, we construct three different datasets: Diversity11, Indo12, and Indo9 which are described in Table 1. To construct the Diversity11 dataset, we randomly select 11 kinds of languages from different districts with varying diversities and quantities, and divide them into 9 source languages and 2 target languages. For Indo12, we randomly select 11 kinds of languages with little difference from the same Indo-European language family and 1 Afro-Asiatic language. To test our method on fewer source languages, we remove 3 languages (Russian, Swedish, and Welsh) from Indo12 to obtain Indo9.

Diversity11	Source	Turkish	13	Tatar	25	Tamil	3
		Swedish	5	Mongolian	9	Latvian	4
		Dhivehi	6	Breton	5	Arabic	7
	Target	Kyrgyz-*train*	10		Kyrgyz-*test*	1	
		Estonian-*train*	9		Estonian-*test*	1	
Indo12	Source	English	10	Portuges	10	Russian	10
		French	10	German	10	Welsh	10
		Italian	10	Catalan	10	Swedish	5
	Target	Spanish-*train*	10		Spanish-*test*	1.5	
		Dutch-*train*	10		Dutch-*test*	1.5	
		Kabyle-*train*	10		Kabyle-*test*	1.5	

Table 1: Multilingual dataset statistics in terms of hours (h).

In addition, we also conducted experiments on the IARPA BABEL dataset (Gales et al. 2014) with 6 source languages (Bengali, Tagalog, Zulu, Turkish, Lithuanian, Guarani) and 3 target languages (Vietnamese, Swahili, Tamil).

Implementation Details. We use the joint attention-CTC ASR model (Kim, Hori, and Watanabe 2017; Hori et al. 2017) as our ASR model. The encoder contains a 6-layered VGG (Simonyan and Zisserman 2015) extractor and 5 BLSTM (Graves, Jaitly, and rahman Mohamed 2013; Graves and Jaitly 2014) layers, each with 320-dimensional units per direction. Location-aware attention (Chorowski et al. 2015) with 300 dimensions is used in our attention layer and the decoder is a single LSTM (Hochreiter and Schmidhuber 1997) layer with 320 dimensions. We set λ_{ctc} to 0.5. During inference, the greedy-search decoding is used to get the best hypothesis. Following (Hori et al. 2017), we use 80-dimensional log Mel-scale filterbank coefficients with pitch features as the input features. Byte pair encoding (BPE) compression algorithm (Sennrich et al. 2016) is employed to process audio transcripts. All transcripts in the different multilingual

Target	Kyrgyz	Estonian	Spanish		Dutch		Kabyle	
Source	*Diversity11*		*Indo9*	*Indo12*	*Indo9*	*Indo12*	*Indo9*	*Indo12*
Monolingual training (Hori et al. 2017)	76.25	86.04	80.30		68.71		85.41	
TL-ASR (Kunze et al. 2017)	68.28	82.04	79.39		56.58		89.12	
MTL-ASR (multi-head) (Dalmia et al. 2018)	67.56	81.50	75.82	73.00	57.80	56.41	82.10	81.23
MTL-ASR (Watanabe, Hori, and Hershey 2017)	64.90	83.70	73.85	71.40	62.55	58.46	84.25	81.88
our AMS (MTL-ASR)	**59.55**	**79.33**	**71.02**	**68.97**	**58.22**	**54.96**	**83.90**	**79.26**
MML-ASR (Hsu, Chen, and yi Lee 2020)	58.29	79.66	66.75	65.24	53.33	52.56	79.45	75.96
our AMS (MML-ASR)	**50.72**	**72.26**	**65.21**	**64.40**	**51.18**	**49.13**	**78.21**	**73.69**

Table 2: Results of low resource ASR on Diversity11, Indo12 and Indo9 in terms of WER (%).

datasets are used to train sub-word models separately based on the BPE algorithm. The transcripts are pre-tokenized to sequences of sub-word units (tokens) one-hot vectors using the sub-word models. The policy network contains a feed forward attention and a one-layer LSTM with the hidden size 100 and the input size 32. We use Adam with an initial learning rate $\gamma = 0.035$ and an entropy penalty weight 10^{-5} to train the policy network. We set M as 3 after searching the range $M \in \{2, 3, 4, 5, 7, 9\}$ and set w as 48, of which 24 examples are divided into support set and 24 examples into query set.

Results on Low-resource ASR

Results on Diversity11. Table 2 reports the results on Diversity11 in terms of word error rate (WER). For all target languages, our AMS significantly outperforms all previous methods. First, the performance of monolingual is poor without the help of source languages. Second, meta-learning decrease WER over 6% thanks to its few-shot learning ability by learning better initialization parameters that enjoy fast adaptation ability. Moreover, by learning to sampling tasks for meta-learning, our AMS further improves the results over 7% on this dataset which has large task imbalance.

Results on Indo12 and Indo9. As shown in Table 2, our method consistently achieves the state-of-the-art performance on Indo12 which eliminates task-quantity imbalance and Indo9 which has much fewer source languages for training. This is because our AMS uses adversarial sampling to select better tasks for effective learning and well overcomes the task-difficulty imbalance issue. The results of Kabyle are much worse than that of Spanish and Dutch because Kabyle is an Afro-Asiatic language and all source languages are Indo-European language, which indicates that source languages from the same language family are more helpful for target languages.

Results on IARPA BABEL. In order to further verify the effectiveness of our AMS, we also conducted experiments to compare with previous works on the IARPA BABEL, which is another public multilingual dataset. Table 3 reports the results on BABEL. In addition to comparing with the baselines above, we also selected the results of recent papers for comparison. As can be observed, our AMS achieves the best results for all target languages, which can improve the performance of both MML-ASR and MTL-ASR with the proposed adversarial meta sampling method. It further demonstrates that the improvement of our AMS can be easily reproduced on different multilingual low-resource ASR datasets.

Method	Vietnamese	Swahili	Tamil
Monolingual (Multi-CTC)(Hsu et al. 2020)	71.80	47.50	69.90
Monolingual (BLSTMP) (Cho et al. 2018)	54.30	33.10	55.30
Monolingual (VGG-Small) (Chen et al. 2020)	46.00	39.60	57.90
Monolingual (VGG-Large) (Chen et al. 2020)	48.30	38.30	60.10
Monolingual (Joint attention-CTC) (Hori et al. 2017)	48.68	38.62	54.45
MTL-ASR (Multi-CTC)(Hsu el al. 2020)	59.70	48.80	65.60
MTL-ASR (Joint attention-CTC)(Watanabe et al. 2017)	47.17	34.10	51.17
our AMS (MTL-ASR)	**45.51**	**33.15**	**49.57**
MML-ASR (Multi-CTC)(Hsu et al. 2020)	50.10	42.90	58.90
MML-ASR (Joint attention-CTC)(Hsu et al. 2020)	45.10	36.14	50.61
our AMS (MML-ASR)	**43.35**	**32.19**	**48.56**

Table 3: Results of low resource ASR on IARPA BABEL in terms of Character Error Rate (CER%).

Method	Kyrgyz	Estonian
MML-ASR (Reptile) (Nichol, Achiam, and Schulman 2018)	66.51	83.17
MML-ASR (FOMAML) (Hsu, Chen, and yi Lee 2020)	59.23	78.64
MML-ASR (MAML) (Uniform) (Hsu, Chen, and yi Lee 2020)	58.29	79.66
PPQ-MAML (Dou, Yu, and Anastasopoulos 2019)	58.95	77.26
PPQL-MAML (Sun et al. 2018a)	54.87	74.97
PPEAQL-MAML	55.14	75.41
PPAQL-MAML	53.15	73.33
our AMS-MAML w/o attention	54.16	74.29
our AMS-Reptile	59.30	78.49
our AMS-FOMAML	53.04	74.77
our AMS-MAML	**50.72**	**72.26**
our AMS-MAML (80% target)	59.02	75.87
our AMS-MAML (50% target)	70.27	81.97
our AMS-MAML (20% target)	87.11	91.72

Table 4: Ablation study results on Diversity11 in terms of WER (%).

Ablation Studies

Considering the superior performance of MML-ASR over MTL-ASR, all the following ablation experiments focus on AMS based on MML-ASR.

Different meta-learning methods. Table 4 shows that our AMS can improve all meta-learning methods, including MAML (Hsu et al. 2020; Finn et al. 2017), FOMAML (Hsu et al. 2020; Finn et al. 2017), Reptile (Nichol et al. 2018). Among them, AMS-FOMAML achieves similar performance as AMS-MAML but has higher training efficiency. So in the other speech tasks, we focus on AMS- FOMAML.

Different scale of training data. To test our method when data are only a few, we reduce the training data of target languages to 80%, 50% and 20%. From Table 4 and 2, one

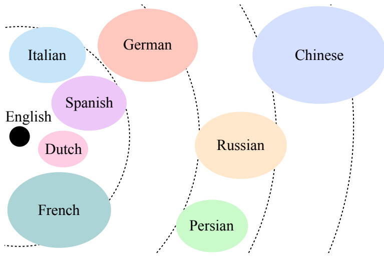


Figure 3: Imbalance analysis of task difficulty in our speech translation experiments. The distance from the dotted circle where the language is located to the black point represents the distance from the language to English and the circular area of the language represents the sample times of this language.

can observe that with 80% training data, our method still works better than most baselines with 100% training data, which testifies our method can effectively alleviate the need of heavy annotated training data.

Comparison among different sampling methods. We further compare the performances of different sampling methods. 1) Sampling tasks uniformly (Uniform). 2) Sampling tasks with the probability proportional to the task quantity of each language task set (PPQ) (Dou, Yu, and Anastasopoulos 2019). 3) Sampling tasks with the probability proportional to the query loss of each language task set (PPQL) (Sun et al. 2018a). 4) Sampling tasks with the probability proportional to the average query loss of each language task set with a window (PPAQL). 5) Sampling tasks with the probability proportional to the exponential average query loss of each language task set (PPEAQL). 6) Our AMS without attention layer (AMS w/o attention). From Table 4, one can find that (1) PPQ is slightly better than Uniform by considering the task-quantity imbalance; (2) the methods that use query loss to sample tasks have greatly improved the performance and PPAQL is the best; (3) our AMS significantly surpasses other methods, which indicates that our policy network can effectively exploit the long-term and instant information to sample the most benefiting tasks in the training process.

Generalization analyses

AMS on multilingual transfer learning. Our sampling method can be generalized to MTL-ASR as mentioned in Sec. . As shown in Table 2, our AMS (MTL-ASR) outperforms MTL-ASR on all datasets. It shows that our AMS can effectively improve both meta-learning methods and multilingual transfer learning methods by simply incorporating a policy network for adversarial sampling.

AMS on speech classification. Our speech classification datasets contain 5 source datasets and 5 target datasets provided by the AutoSpeech 2020 competition[1]. Different datasets come from different speech classification domains with varying examples, classes and the quantity of examples, including speaker identification, emotion classification, etc. We evaluate our AMS with MobileNetV2 (Sandler et al. 2018) as the feature extractor and NetVLAD (Arandjelović et al. 2016) as the aggregation network. As shown in Table 5, our AMS outperforms most of the baselines with large improvement.

[1]https://www.automl.ai/competitions/2

Method	*D1*	*D2*	*D3*	*D4*	*D5*	Avg acc
Train from scratch	3.53	81.32	36.08	47.0	3.35	34.26
Pretraining	4.77	78.0	41.09	48.45	3.93	35.25
FOMAML (Finn et al. 2017)	9.8	77.8	42.27	**49.76**	6.39	37.20
AMS-FOMAML	**10.8**	**82.36**	**45.70**	49.09	**10.21**	**39.63**

Table 5: Results of speech classification in terms of accuracy(%).

Method	Mongolian	Swedish	Turkish
CoVoST scratch (Wang et al. 2020)	0.20	0.30	0.80
Train from scratch	0.27	0.22	0.85
Pretraining	0.30	0.57	1.26
FOMAML (Finn et al. 2017)	0.35	0.67	1.26
AMS-FOMAML	**0.36**	**0.70**	**1.45**

Table 6: Results of speech translation in terms of BLEU.

AMS on speech translation. Here we consider translating other languages speech to English. We select 8 source languages of 10 hours and 3 target languages less than 10 hours from CoVoST (Wang et al. 2020), a multilingual speech translation (ST) corpus. For simplicity, we use the same model architecture and data preprocessing procedure as ASR in Sec. , which can achieve the same performance as the model used in CoVoST. Table 6 shows the result of case-insensitive tokenized BLEU (Papineni et al. 2002) using sacreBLEU (Post 2018). By comparison, our method outperforms all baselines, including CoVoST, and achieves state-of-the-art performance in all the three target languages.

Analysis of difficulty imbalance. We limit the quantity of each source language to 10 hours to analyze the task difficulty in our ST experiments. In Figure 3, we use the semantic similarity between English and other source languages (Senel et al. 2018) as well as the language learning difficulty for English speakers (FSI. 2007) as references to measure the distances, where the farther the language is, the more difficult is to translate it into English. It can be observed a trend that the sample times increase with the distances. For example, Chinese and Dutch are the farthest language with the highest sampling times and the closest language with the fewest sample times, respectively. This indicates that our method can automatically sample tasks according to the task difficulty to alleviate the imbalance from different language difficulties.

Conclusion

In this work, to tackle the task-imbalance problem caused by language tasks difficulties and quantities, we develop a novel Adversarial Meta Sampling framework to adaptively sample language tasks for learning a better model initialization for target low-resource languages. It can well handle the challenging multilingual low-resource ASR in real world. Extensive experimental results validate that our method effectively improves the few-shot learning ability of both meta-learning and transfer learning and also shows its great generalization capacity in other low-resource speech tasks.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China (NSFC) under Grant No.U19A2073 and No.61976233, Guangdong Province Basic and Applied Basic Research (Regional Joint Fund-Key) Grant No.2019B1515120039, Nature Science Foundation of Shenzhen Under Grant No. 2019191361, Zhijiang Lab's Open Fund (No. 2020AA3AB14) and CSIG Young Fellow Support Fund.

References

Adams, O.; Wiesner, M.; Watanabe, S.; and Yarowsky, D. 2019. Massively Multilingual Adversarial Speech Recognition. *NAACL-HLT* 96–108.

Arandjelović, R.; Gronát, P.; Torii, A.; Pajdla, T.; and Sivic, J. 2016. NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 5297–5307.

Ardila, R.; Branson, M.; Davis, K.; Henretty, M.; Kohler, M.; Meyer, J.; Morais, R.; Saunders, L.; Tyers, F. M.; and Weber, G. 2020. Common Voice: A Massively-Multilingual Speech Corpus. In *LREC*.

Chan, W.; Jaitly, N.; Le, Q. V.; and Vinyals, O. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 4960–4964.

Chen, Y.-C.; Hsu, J.-Y.; Lee, C.-W.; and yi Lee, H. 2020. DARTS-ASR: Differentiable Architecture Search for Multilingual Speech Recognition and Adaptation. In *INTERSPEECH*.

Cho, J.; Baskar, M. K.; Li, R.; Wiesner, M.; Mallidi, S. H.; Yalta, N.; Karafiát, M.; Watanabe, S.; and Hori, T. 2018. Multilingual Sequence-to-Sequence Speech Recognition: Architecture, Transfer Learning, and Language Modeling. *2018 IEEE Spoken Language Technology Workshop (SLT)* 521–527.

Chorowski, J.; Bahdanau, D.; Serdyuk, D.; Cho, K.; and Bengio, Y. 2015. Attention-Based Models for Speech Recognition. In *NIPS*.

Chung, Y.-A.; and Glass, J. 2020. Generative Pre-Training for Speech with Autoregressive Predictive Coding. *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* .

Dalmia, S.; Sanabria, R.; Metze, F.; and Black, A. W. 2018. Sequence-Based Multi-Lingual Low Resource Speech Recognition. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 4909–4913.

Dou, Z.-Y.; Yu, K.; and Anastasopoulos, A. 2019. Investigating Meta-Learning Algorithms for Low-Resource Natural Language Understanding Tasks. In *EMNLP/IJCNLP*.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *ICML*.

FSI. 2007. Language Learning Difficulty for English Speakers. https://en.wikibooks.org/wiki/Wikibooks:Language_Learning_Difficulty_for_English_Speakers.

Gales, M.; Knill, K.; Ragni, A.; and Rath, S. P. 2014. Speech recognition and keyword spotting for low-resource languages: Babel project research at CUED. In *SLTU*.

Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; and Lempitsky, V. S. 2016. Domain-Adversarial Training of Neural Networks. *Journal of Machine Learning Research* vol. 17, no. 1, pp. 2096–2030.

Graves, A.; Fernández, S.; Gomez, F. J.; and Schmidhuber, J. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *ICML '06*.

Graves, A.; and Jaitly, N. 2014. Towards End-To-End Speech Recognition with Recurrent Neural Networks. In *ICML*.

Graves, A.; Jaitly, N.; and rahman Mohamed, A. 2013. Hybrid speech recognition with Deep Bidirectional LSTM. *2013 IEEE Workshop on Automatic Speech Recognition and Understanding* 273–278.

Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Computation* 9: 1735–1780.

Hori, T.; Watanabe, S.; Zhang, Y. L.; and Chan, W. 2017. Advances in Joint CTC-Attention based End-to-End Speech Recognition with a Deep CNN Encoder and RNN-LM. In *INTERSPEECH*.

Hsu, J.-Y.; Chen, Y.-J.; and yi Lee, H. 2020. Meta Learning for End-to-End Low-Resource Speech Recognition. *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* .

Hu, K.; Bruguier, A.; Sainath, T. N.; Prabhavalkar, R.; and Pundak, G. 2019. Phoneme-Based Contextualization for Cross-Lingual Speech Recognition in End-to-End Models. In *Proc. Interspeech 2019*, 2155–2159.

Kahn, J.; Lee, A.; and Hannun, A. 2020. Self-Training for End-to-End Speech Recognition. *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* .

Kim, S.; Hori, T.; and Watanabe, S. 2017. Joint CTC-attention based end-to-end speech recognition using multi-task learning. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 4835–4839.

Kunze, J.; Kirsch, L.; Kurenkov, I.; Krug, A.; Johannsmeier, J.; and Stober, S. 2017. Transfer Learning for Speech Recognition on a Budget. In *Rep4NLP,ACL*.

Li, B.; Sainath, T. N.; Pang, R.; and Wu, Z. 2019. Semi-supervised Training for End-to-end Models via Weak Distillation. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 2837–2841.

Nichol, A.; Achiam, J.; and Schulman, J. 2018. On First-Order Meta-Learning Algorithms. *ArXiv* abs/1803.02999.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *ACL*.

Post, M. 2018. A Call for Clarity in Reporting BLEU Scores. In *WMT*.

Pratap, V.; Hannun, A.; Xu, Q.; Cai, J.; Kahn, J.; Synnaeve, G.; Liptchinsky, V.; and Collobert, R. 2019. Wav2Letter++: A Fast Open-source Speech Recognition System. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 6460–6464.

Sandler, M.; Howard, A. G.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C. 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* 4510–4520.

Schneider, S.; Baevski, A.; Collobert, R.; and Auli, M. 2019. wav2vec: Unsupervised Pre-training for Speech Recognition. In *INTERSPEECH 2019*.

Senel, L. K.; Utlu, I.; Yücesoy, V.; Koc, A.; and Çukur, T. 2018. Generating Semantic Similarity Atlas for Natural Languages. *2018 IEEE Spoken Language Technology Workshop (SLT)* 795–799.

Shinohara, Y. 2016. Adversarial Multi-Task Learning of Deep Neural Networks for Robust Speech Recognition. In *INTERSPEECH*.

Simonyan, K.; and Zisserman, A. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* abs/1409.1556.

Sun, Q.; Liu, Y.; Chua, T.-S.; and Schiele, B. 2018a. Meta-Transfer Learning for Few-Shot Learning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 403–412.

Sun, S.; Yeh, C.-F.; Hwang, M.-Y.; Ostendorf, M.; and Xie, L. 2018b. Domain Adversarial Training for Accented Speech Recognition. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 4854–4858.

Tong, S.; Garner, P. N.; and Bourlard, H. 2017. Multilingual Training and Cross-lingual Adaptation on CTC-based Acoustic Model. *ArXiv* abs/1711.10025.

Toshniwal, S.; Sainath, T. N.; Weiss, R. J.; Li, B.; Moreno, P. J.; Weinstein, E.; and Rao, K. 2018. Multilingual Speech Recognition with a Single End-to-End Model. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 4904–4908.

Waibel, A.; Soltau, H.; Schultz, T.; Schaaf, T.; and Metze, F. 2000. *Multilingual Speech Recognition*, 33–45. Springer Berlin Heidelberg.

Wang, C.; Pino, J.; Wu, A.; and Gu, J. 2020. CoVoST: A Diverse Multilingual Speech-To-Text Translation Corpus. *ArXiv* abs/2002.01320.

Wang, X.; Tsvetkov, Y.; and Neubig, G. 2020. Balancing Training for Multilingual Neural Machine Translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 8526–8537.

Watanabe, S.; Hori, T.; and Hershey, J. R. 2017. Language independent end-to-end architecture for joint language identification and speech recognition. *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)* 265–271.

Williams, R. J. 1992. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning* 8: 229–256.

Winata, G. I.; Cahyawijaya, S.; Lin, Z.; Liu, Z.; Xu, P.; and Fung, P. 2020. Meta-Transfer Learning for Code-Switched Speech Recognition. In *ACL*.

Yi, J.; Tao, J.; Wen, Z.; and Bai, Y. 2018. Adversarial Multilingual Training for Low-Resource Speech Recognition. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 4899–4903.

Zhou, P.; Yuan, X.; Xu, H.; Yan, S.; and Feng, J. 2019. Efficient Meta Learning via Minibatch Proximal Update. In *NeurIPS*.

Zhou, P.; Zou, Y.; Yuan, X.; Feng, J.; Xiong, C.; and Hoi, S. C. 2020. Task Similarity Aware Meta Learning: Theory-inspired Improvement on MAML. In *4th Workshop on Meta-Learning at NeurIPS*.