ADAPT-AND-ADJUST: OVERCOMING THE LONG-TAIL PROBLEM OF MULTILINGUAL SPEECH RECOGNITION

Anonymous authors
Paper under double-blind review

ABSTRACT

One crucial challenge of real-world multilingual speech recognition is the long-tailed distribution problem, where some resource-rich languages like English have abundant training data, but a long tail of low-resource languages have varying amounts of limited training data. To overcome the long-tail problem, in this paper, we propose the Adapt-and-Adjust (A2), a transformer-based multi-task learning framework for end-to-end multilingual speech recognition. The A2 framework overcomes the long-tail problem via three types of techniques: (1) exploiting a pretrained multilingual language model (mBERT) to improve the performance of low-resource languages; (2) proposing dual adapters consisting of both language-specific and language-agnostic adaptation with minimal additional parameters; (3) overcoming the class imbalance either by imposing class priors in the loss during training or adjusting the logits of softmax output during inference. Extensive experiments on the CommonVoice corpus show that A2 significantly outperforms conventional approaches for end-to-end multilingual speech recognition.

1 INTRODUCTION

Deploying a single Automatic Speech Recognition (ASR) model to recognize multiple languages is highly desired but very challenging for real-world multilingual ASR scenarios due to the well-known long-tailed distribution challenge, i.e., some resource-rich languages like English have abundant training data while a long tail of low-resource languages have limited varying amounts of training data. The recent popular end-to-end (E2E) monolingual ASR architecture (Graves et al., 2013; Chan et al., 2015; Vaswani et al., 2017) is promising for achieving the state-of-the-art performance for resource-rich languages, but suffers dramatically from the long tail of low-resource languages due to the lack of training data. In this paper, we aim to investigate an end-to-end multilingual ASR framework where a single model is trained end-to-end from a pooled dataset of all target languages to improve the overall performance of multilingual ASR tasks especially for low-resource languages.

While a standard end-to-end multilingual multi-task training approach is able to improve the overall performance as compared with the monolingual end-to-end approaches, it is not able to fully overcome the long-tail problem. One of key reasons is due to the class imbalance issue, which will make the multilingual model bias towards the dominant languages. To address the class imbalance issue, one straightforward approach is to resample the training data (Kannan et al., 2019; Pratap et al., 2020) during batch assemble. However, such an ad-hoc approach does not fully resolve the underlying long-tail distribution problem, and only a marginal improvement is obtained in practice.

In this paper, we propose the Adapt-and-Adjust (A2) framework for multilingual speech recognition using a speech transformer to address the long-tail data distribution issue. For adapting the multilingual network to specific languages, both language-specific and language-agnostic adapters are augmented to each encoder and decoder layer for adaptation of the multilingual model to each language with minimal additional parameters. While the language-specific adapters focus on adapting the shared network weights to a particular language, a common adapter is proposed to learn some shared and language-agnostic knowledge for better knowledge transfer across languages. Moreover, to increase the relative margin between logits of rare versus and dominant languages, we perform class imbalance adjustments during multilingual model training or inference by revisiting the classic idea of logit adjustment (Zhou & Liu, 2006; Menon et al., 2020). Class imbalance adjustment (Collell et al., 2016; Menon et al., 2020; Cui et al., 2019) is applied by adjusting the logits of

softmax input with the class priors. Lastly, for better language modeling, a distilled mBERT (Devlin et al., 2019) is converted to an autoregressive transformer decoder to jointly explore the multilingual acoustic and text space to further improve the performance of low-resource languages.

We conduct experiments and establish a benchmark from the CommonVoice corpus with a realistic long-tailed distribution of different languages. Extensive experiments show that A2 significantly outperforms conventional approaches for end-to-end multilingual ASR. Our key contributions are:

- We propose the Adapt-and-Adjust (A2), a novel end-to-end transformer-based framework for real-world multilingual speech recognition to overcome the long-tail challenge;
- We demonstrate the effectiveness of utilizing a pretrained multilingual language model as a speech decoder to improve multilingual text representations and language adapters to better share the learned information across all languages. To the best of our knowledge, this work is the first to adapt a pretrained multilingual language model for multilingual ASR.
- We show that the incorporation of class priors during training or inference is effective and essential to address the long-tail issue in multilingual training.
- We establish a reproducible multilingual speech recognition benchmark with long-tailed distributions of 11 languages from different language families for the research community.

We will release the code of our implementations and the data sets as a benchmark to facilitate the research on addressing the long-tail problem of multilingual ASR.

2 Adapt-and-Adjust Framework

2.1 Overview

Figure 1 gives an overview of the proposed Adapt-and-Adjust framework for end-to-end multilingual ASR tasks. Our framework improves a basic transformer-based sequence-to-sequence model by proposing three key novel contributions: (1) mBERT based decoder, (2) language adapters, and (3) class-imbalance adjustments. First of all, we extend the basic transformer framework by replacing the vanilla decoder with an mBERT model for better language modeling, which is particularly important for low-resource languages. Second, the common and language-specific adapters are added to each encoder and decoder layer to learn both the shared and language-specific information for better adaptation capability. Finally, we perform class imbalance adjustments during training or inference phase, in which the raw logits to the softmax layer are adjusted with the class priors estimated from the training data. From a Bayesian point of view, the classifier after the logit adjustment minimizes a balanced error where both the head and tail classes contribute equally to the training.

2.2 Base Model: Hybrid CTC-Attention Speech Transformer

A sequence-to-sequence speech transformer model (Dong et al., 2018; Kim et al., 2016; Karita et al., 2019b) based on the hybrid CTC-Attention network is used for acoustic modeling. It takes in the acoustic features $\mathbf{x} \in \mathbb{R}^{T,F}$ and outputs the sentence piece tokens $\mathbf{y}$, where T, F denote sequence length and feature dimension, respectively. The encoder consists of several 2D convolution layers followed by self-attention layers. The convolution layers are used to extract more robust features before they are sent to the transformer. The decoder layers have two attention mechanisms, one for self-attention and the other one for the encoder output. The network is trained in an autoregressive manner by predicting the next token given the current output. In addition to the attention loss, a connectionist temporal classification (CTC) (Graves et al., 2006) layer is added to the encoder output to serve as a regularizer to the attention model.

Training Multi-task loss $\mathcal{L}_{MTL}$ (Watanabe et al., 2018; Karita et al., 2019b) combining the CTC loss (Graves et al., 2006) and attention loss $\mathcal{L}_{\text{ATTN}}$ is used to train the speech transformer. The multi-task loss is computed as an interpolation of two losses with a hyper-parameter λ ($0 \leq \lambda \leq 1$):

$$\mathcal{L}_{\text{ATTN}} = \text{KL}(\mathbf{y}_{\text{ATTN}} || \mathbf{y}), \tag{1}$$

$$\mathcal{L}_{\text{MTL}} = \lambda \log P_{\text{CTC}}(\mathbf{y}|\mathbf{h}_{enc}) + (1-\lambda)\mathcal{L}_{\text{ATTN}}, \tag{2}$$

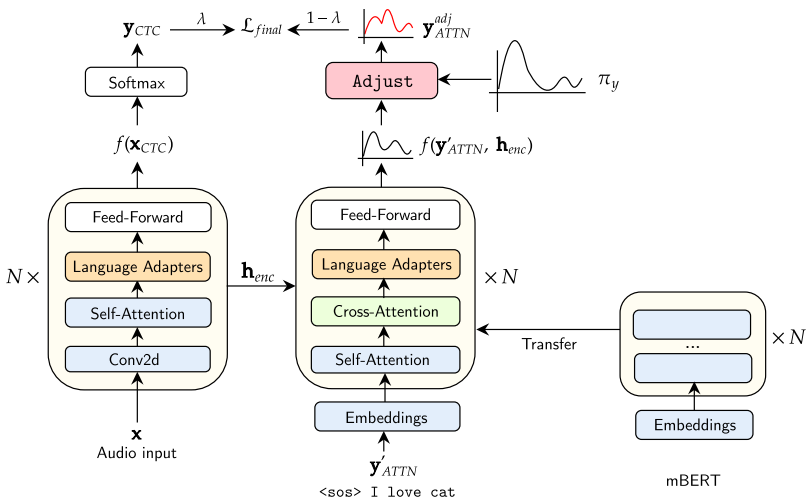

Figure 1: Overview of the Adapt-and-Adjust (A2) framework. Layer norm is omitted to save space. $\mathbf{y}_{\text{CTC}}$ is the output of the connectionist temporal classification (CTC) branch, $\mathbf{y}_{\text{ATTN}}$ is the output of the decoder, and $\mathbf{y}'_{\text{ATTN}}$ is the previous tokens. Three key components are: (1) a pre-trained language model mBERT based decoder to improve the performance of low-resource languages; (2) language adaptation via the proposed dual adapters; and (3) class-imbalance adjustment either during training or inference by adjusting the logits $f(\mathbf{y}'_{\text{ATTN}}, \mathbf{h}_{\text{enc}})$ with class priors π_y.

where $\mathbf{y}$ is the label distribution after label smoothing (Müller et al., 2019) to prevent the model from making over-confident predictions. Kullback-Leibler Divergence loss (KL) (Kullback & Leibler, 1951) instead of cross-entropy is used for the attention loss. The probability distribution $P_{\text{CTC}}(\mathbf{y}|\mathbf{h}_{enc})$ is computed by using dynamic programming.

Decoding Beam search is used to predict the sentence pieces without any additional language models. The decoding score is computed as a weighted sum of both the CTC and attention network probabilities by using β as the decoding parameter to balance them (Karita et al., 2019a).

$$\hat{\mathbf{y}} = \underset{\mathbf{y}\in\mathcal{Y}^*}{\arg\max}\{\beta P_{\text{CTC}}(\mathbf{y}|\mathbf{h}_{\text{enc}}) + (1-\beta)P_{\text{ATTN}}(\mathbf{y}|\mathbf{h}_{\text{enc}}, \mathbf{y}')\}, \tag{3}$$

where $\mathbf{y}'$ is the decoded sequence so far.

2.3 Multilingual BERT as Transformer Decoder

For better language modeling, especially for low-resource languages, mBERT is used as the transformer decoder. Since mBERT was initially trained on text data, to jointly explore the acoustic and text space, it is essential to augment a cross-attention layer to the encoder output for each mBERT layer in addition to its self-attention layer to "align" the two spaces for the speech recognition task. This is because the text space may diverge significantly from the acoustic space of the encoder output. Another change is that the mBERT is set to be autoregressive: future positions are masked for prediction, given the current output token and the encoder output.

Autoregressive mBERT Figure 7 depicts the changes made to the vanilla mBERT so that it can be used as the speech transformer decoder. Let t denote the current time step. The autoregressive decoder takes the current input token y_t to predict the next token y_{t+1}. The mBERT embedding layer converts the input token y_t to a vector representation $\mathbf{v}_t$, which is used as the key, value, and query of the self-attention layer. Subsequently, the cross-attention layer takes the encoder output $\mathbf{h}_{enc}$ as the key and value, the self-attention output as the query, and compute the attention output.

Vocabulary Mapping The vocabulary size of the vanilla mBERT is too large (119,547 tokens) for training the end-to-end speech recognition system. Therefore, vocabulary mapping is performed to reduce the number of targets for the speech transformer. In this work, sentence pieces (SP) (Kudo,

2018) are used as the target tokens for all the end-to-end ASR systems. The SP models are trained on the transcriptions with a preset number of vocabulary size. In this work, we use a shared set of 5,237 tokens as the vocabulary for the multilingual system. The minimum number of the token set for the sentence piece model is 150 for all the monolingual systems, except Chinese with 2,265 tokens. The generated sentence piece tokens are then matched against the mBERT token set. During training, the embeddings of all tokens in the mBERT vocabulary are initialized with mBERT embeddings. The vocabulary coverage is shown in Appendix Table 8.

2.4 Language Adapters

Similar to Kannan et al. (2019), lightweight residual language adapters are used to mitigate the data imbalance issues. As shown in Figure 2, in addition to the language-specific adapter for capturing the language-intrinsic knowledge, a common shared adapter is also trained to learn language-agnostic information in the multilingual data, we call this **Dual-Adapters**. The language-specific and common adapters are denoted as A_{lang} and A_{com} respectively. Each adapter of layer l consists of a down-projection layer $\mathbf{W}_d^l$, followed by a ReLU activation function, and an up-projection layer $\mathbf{W}_u^l$. The adapters take $\mathbf{h}^l$ as the input, where $\mathbf{h}^l$ is the self attention output of layer l. We compute the output of the adapter $\text{Adapter}(\mathbf{h}^l)$ as following for both language-specific and common adapters:

$$\text{Adapter}(\mathbf{h}^l) = \mathbf{W}_u^l(\text{ReLU}(\mathbf{W}_d^l(\text{LayerNorm}(\mathbf{h}^l)))) + \mathbf{h}^l. \tag{4}$$

Afterwards, the final adapter output $\mathbf{o}^l = \mathbf{o}_{\text{lang}}^l + \mathbf{o}_{\text{com}}^l$. Then, $\mathbf{o}^l$ is used as the input to the next encoder or decoder layer. We create a language mask to specify the language-specific adapters.

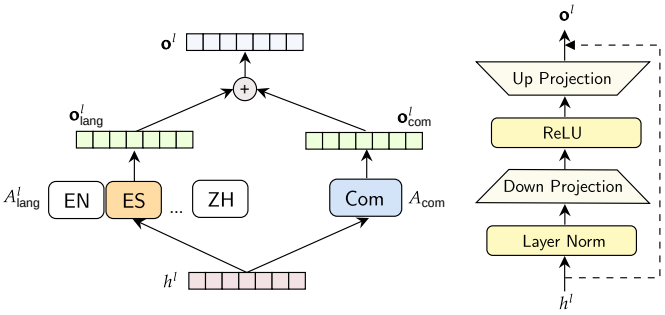

Figure 2: Dual-Adapters. The orange box presents the active language-specific adapter and the blue box presents the common adapter. The figure on the right shows the language adapter structure.

2.5 Class Imbalance Adjustments

In addition to the architecture extension with language adapters, the class imbalance problem is addressed to incorporate the class priors during training or inference via logit adjustments. Derived from a Bayesian classifier point of view in Menon et al. (2020) for computer vision tasks, the softmax classifier with adjusted logits as input minimizes the balanced error across all classes. A natural adjustment is to scale the raw logits $f_y(x)$ by the inverse of its class prior π_y. In log domain, the logit adjustment can be performed as following:

$$f_y^{\text{adj}}(x) = f_y(x) - \tau \cdot \log \pi_y \tag{5}$$

where $\tau > 0$ is a hyper-parameter. The adjustment can be viewed as applying a class-dependent offset to re-weight each logit according to its class prior. The logit adjustment can be performed either during both model training or inference.

Class priors The class priors are the natural frequencies of the sentence piece tokens estimated from the multilingual training data. To form a valid prior distribution, smoothing is applied to the raw counts according to Equation 6 for zero occurrence tokens:

$$\pi_y = \begin{cases} c_i/C - 1/(n_0 * C), & c_i > 0 \\ 1/C, & \text{otherwise,} \end{cases} \tag{6}$$

where C is the total number of counts for all labels, n_0 is the number of labels with zero occurrences, N is as the number of classes and c_i is the raw count of class i.

Training phase class imbalance adjustments To incorporate the priors during training, the logits $f_y^{\text{dec}}(t)$ of the last decoder layer are adjusted before softmax according to the following:

$$f_y^{\text{dec}}(t) = w_y^T \cdot \text{Decoder}(\mathbf{h}_{\text{enc}}, \text{Embedding}(y_{t-1})) \tag{7}$$

$$f_y^{\text{adj}}(t) = f_y^{\text{dec}}(t) - \tau \cdot \log \pi_{y_t}, \tag{8}$$

$$\mathbf{y}_t^{\text{adj}} = \frac{\exp(f_y^{\text{adj}}(t))}{\sum_{y' \in [N]} \exp(f_{y'}^{\text{adj}}(t))} \tag{9}$$

The adjusted softmax outputs $\mathbf{y}^{\text{adj}}$ of the sequence are used to compute the KL loss and perform the backward propagation to update the model. y_{t-1} is the previous label available only during training. To reduce the training and inference discrepancy, scheduled sampling (Bengio et al., 2015) is commonly used for the sequential classification tasks like speech recognition. During later training iterations, instead of using the ground truth label y_{t-1} for computing the logits, y'_{t-1} is chosen from the maximum prediction output of the current model to simulate the inference:

$$y'_{t-1} = \underset{y}{\text{argmax}}\, \mathbf{y}_{t-1}^{\text{adj}} \tag{10}$$

If scheduled sampling is used, the adjusted logits at step t will have influence over all following tokens in the current sequence. This is a crucial difference from the image classification task in Menon et al. (2020).

Inference phase class imbalance adjustments Alternatively, the class priors can be incorporated during inference via logit adjustments. The decoding score is computed as following:

$$\hat{\mathbf{y}} = \underset{\mathbf{y} \in \mathcal{Y}^*}{\arg\max} \{\beta P_{\text{CTC}}(\mathbf{y}|\mathbf{h}_{\text{enc}}) + (1-\beta)\mathbf{y}^{\text{adj}}\} \tag{11}$$

During beam search, the attention decoding scores y_t^{adj} are computed in the same way as the scheduled sampling from the adjusted logits.

3 EXPERIMENTS

3.1 EXPERIMENTAL SETUP

Dataset The CommonVoice dataset (Ardila et al., 2020) is a multilingual corpus collected by Mozilla. Similar to (Conneau et al., 2020), we use 11 languages: *English (en), Spanish (es), French (fr), Italian (it), Kyrgyz (ky), Dutch (nl), Russian (ru), Swedish (sv), Turkish (tr), Tatar (tt),* and *Chinese (zh)*. The dataset is split into training, dev, and eval set according to the ESPnet recipe.[1] We then sample the split dataset for generating the long-tailed distribution. The acoustic features are 80-dimensional filter banks with pitch. The transcriptions are tokenized using the SentencePiece model with the unigram algorithm. We train our SentencePiece model using our transcription data, and the size of the vocabulary is shown in Table 8. We then add special tokens, such as `<unk>`, `<sos>`, `<eos>`, and a blank token for the CTC objective.

Network configurations We use six transformer encoder layers with a hidden size of 2048 units and eight attention heads, each with attention dim of 256. For the decoder, distil-mBERT (Sanh et al., 2019) is used. The mBERT decoder consists of six transformer decoder layers with a hidden size of 3072 and attention dim of 756 and four attention heads. We train our model with a batch size of 32 and 2 times gradient accumulation in a single GPU NVIDIA V100 16GB with Adam optimizer with a warm-up step of 25000. Particularly, for the balance sampling, we take 6 samples for each language and construct a balanced batch by accumulating gradient for 11 times. Gradient clipping of 5 is applied to stabilize the training. The details of the split is shown in Table 6.

[1] https://github.com/espnet/espnet/tree/master/egs/commonvoice

Training and Evaluation We evaluate our model using beam-search with a beamwidth of 10 and $\lambda = 0.3$ and $\beta = 0.5$. The hyperparameter τ is set to 0.3 for both training and inference phase class imbalance adjustments. The multilingual models are trained with 150K iterations. We compute the average over the last ten checkpoints as the decoding model. For the monolingual setting, we stop after 100 epochs training. Models are evaluated using character error rate (CER) to simplify the evaluation and have a universal metric for all languages.

Baselines As baseline approaches, we consider **Monolingual**: we train models on single language; **MT**: Standard Multilingual Training, we randomly sample the batch from the data distribution; **BS**: Balanced sampling, we sample the same number of utterances for each language in a batch so that they have roughly equal contributions to the training; and **LAN-Specific Adapters**: language-specific adapters proposed by Kannan et al. (2019).

3.2 RESULTS

In Table 1, we present the test results on the CommonVoice dataset. Compared to the monolingual models, even MT models improve the performance of the low-resource languages significantly. This is due to the human languages share some common sub-phonetic articulatory features (Wang & Sim, 2014) even though their phoneme set may vary. This shared knowledge may be captured in the multilingual model, which helps the recognition of low-resource languages.

Balanced sampling We observe the same trend as in Kannan et al. (2019): compared to the MT, the tail classes performance has been significantly boosted. However, the head classes suffer from the balanced sampling due to fewer occurrences during training and possibly model overfitting to the tail classes. Consequently, the overall performance is the same as MT. In fact, even after balanced sampling, the sentence piece tokens still have a long-tailed distribution as shown in Appendix.

Table 1: Test results in CER (%) on the CommonVoice dataset.

	en	**fr**	**es**	**it**	**ru**	**zh**	**tt**	**nl**	**tr**	**ky**	**sv**	**avg**
Train hour	80	50	40	20	15	15	15	15	10	9	4	
Monolingual	**13.3**	**11.5**	**11.1**	20.7	**10.3**	**28.2**	13.6	23.0	28.0	30.0	56.1	22.3
Standard Multilingual Training (MT)	20.1	17.4	**13.0**	**12.5**	13.7	34.1	**12.2**	**18.7**	**13.9**	**14.4**	**26.4**	**17.9**
Balanced Sampling (BS)	25.2	20.3	14.5	12.7	13.2	32.6	11.5	18.0	12.9	13.4	22.8	17.9
LAN-Specific Adapters	**24.2**	**19.4**	**13.9**	**12.3**	**11.8**	**32.0**	**10.7**	**16.9**	**11.8**	**12.7**	**21.7**	**17.0**
A2 (Adjust-Inference)	22.6	17.9	12.7	**11.2**	11.2	**30.1**	10.2	**15.8**	11.4	12.2	**21.1**	**16.0**
A2 (Adjust-Train)	**22.0**	**17.7**	**12.5**	11.3	**11.1**	30.4	**10.0**	15.9	**11.3**	**12.1**	21.3	**16.0**

Language Adapters We first only use the language-specific adapters (LAN-Specific Adapters), then we compare the results with Dual-Adapters. Note that all adapters are based on the BS + mBERT, which has a better performance than the BS only model. We study the impact of mBERT in a separate part. Different from findings in Kannan et al. (2019), where a marginal 0.2% performance gain is obtained with language-specific adapters, adding adapters helps all the languages. Adding the language-specific adapters as in Kannan et al. (2019) without common adapters significantly outperform the balanced sampling baseline with 0.9% absolute performance gain.

Class Imbalance Adjustment In addition to adapters, the long-tail distribution issue is further addressed with class imbalance adjustments. Both the training and inference phase adjustments provide a significant performance gain over the LAN-Specific Adapters with 1% absolute CER reduction. The gains are mostly due to the improved performance of the head languages, although tail languages also benefit from the logit adjustments. More importantly, the gap between the monolingual and multilingual performance for the head classes is greatly reduced, leading to a better "balanced error" performance. This strongly justifies the importance of class imbalance adjustments.

3.3 ABLATION STUDIES

3.3.1 MULTILINGUAL BERT

The effectiveness of mBERT is presented in Table 2. The performance of mBERT depends heavily on the quality of acoustic models. Without adapters and logit adjustments, the improvement over BS is marginal and mBERT performance is even worse if standard multilingual training is used. This may indicate that with better acosutic models like A2, the text space of the vanilla mBERT is better aligned with the acoustic space, which leads to improved recognition performance across all languages, especially for low-resource languages. It is also interesting to note that even without adapters or balanced sampling, the "MT + mBERT + Adjust-Train" system yields the same overall performance as the best adapter system (BS + mBERT + Dual-Adapters).

Table 2: Ablation Study on mBERT for multilingual speech recognition

	en	**fr**	**es**	**it**	**ru**	**zh**	**tt**	**nl**	**tr**	**ky**	**sv**	**avg**
MT	**20.1**	**17.4**	**13.0**	**12.5**	**13.7**	**34.1**	**12.2**	**18.7**	**13.9**	14.4	26.4	**17.9**
MT + mBERT	20.4	17.9	13.4	13.0	14.0	34.8	12.4	18.9	14.0	**14.3**	**26.2**	18.1
BS	25.2	20.3	**14.5**	12.7	**13.2**	**32.6**	11.5	18.0	12.9	13.4	**22.8**	17.9
BS + mBERT	**25.0**	**20.0**	15.4	**12.6**	**13.2**	32.9	**11.1**	**17.3**	**12.6**	**12.8**	**22.8**	**17.8**
BS + Dual-Adapters	23.5	18.9	13.5	12.1	12.3	31.0	10.9	16.5	12.0	12.9	21.6	16.8
BS + Dual-Adapters + mBERT	**23.4**	**18.6**	**13.4**	**11.8**	**11.7**	**30.8**	**10.8**	**16.2**	**11.6**	**12.4**	**21.5**	**16.5**
MT + Adjust-Train	20.2	16.9	12.5	11.9	13.1	32.9	11.3	18.5	13.5	13.9	25.3	17.3
MT + Adjust-Train + mBERT	**19.4**	**16.5**	**12.1**	**11.7**	**12.2**	**31.1**	**11.0**	**17.5**	**12.7**	**13.2**	**24.6**	**16.5**

3.3.2 LANGUAGE ADAPTERS

In this section, we investigate the impacts of different language adapter configurations. The results and parameter sizes are given in Table 3. Generally speaking, adding adapters in the decoder layers is not as effective as in the encoder layers, indicating adaption of acoustic space is much more effective than the text space. Therefore, considering the extra computation and parameters, it is advisable to apply only the encoder adapters with only 2M increase in parameters.

Table 3: Ablation Study on language adapters impacts in BS + mBERT models

	Params	**en**	**fr**	**es**	**it**	**ru**	**zh**	**tt**	**nl**	**tr**	**ky**	**sv**	**avg**
No Adapter	76M	25.0	20.0	15.4	12.6	13.2	32.9	11.1	17.3	12.6	12.8	22.8	17.8
Decoder Adapter Only	82M	23.6	19.2	13.7	12.2	11.7	32.5	10.3	16.7	11.7	12.2	21.4	16.8
Encoder Adapter Only	78M	**23.1**	**18.5**	**13.3**	**11.8**	**11.7**	31.1	**10.6**	**16.0**	**11.5**	12.6	**21.3**	**16.5**
Encoder + Decoder	84M	23.4	18.6	13.4	**11.8**	**11.7**	**30.8**	10.8	16.2	11.6	**12.4**	21.5	**16.5**

We then investigate the effectiveness of the common language adapters as shown in Table 4. The Dual-Adapters outperforms the language specific adapters significantly by a 0.5% absolute CER reduction, indicating the knowledge transfer with common adapter is effective.

Table 4: Ablation Study of common and language-specific adapters.

	en	**fr**	**es**	**it**	**ru**	**zh**	**tt**	**nl**	**tr**	**ky**	**sv**	**avg**
LAN-Specific Adapters	24.2	19.4	13.9	12.3	11.8	32.0	10.7	16.9	11.8	12.7	21.7	17.0
Dual-Adapters	**23.4**	**18.6**	**13.4**	**11.8**	**11.7**	**30.8**	**10.8**	**16.2**	**11.6**	**12.4**	**21.5**	**16.5**

3.3.3 CLASS-IMBALANCE ADJUSTMENTS: TRAINING VS. INFERENCE

The two logit adjustment variants are compared with different network configurations in Table 5. For the standard multilingual training system, the training phase adjustment shows a clear advantage over the inference phase adjustment. Under the convex assumption, the solution of the two

adjustment approaches is the same. However, deep neural network optimization is a non-convex problem, they may converge to different local minima. Therefore, for the standard training, the model is heavily biased towards the head classes. Training phase class imbalance adjustment can help the training more focus on the tail classes, leading to much balanced and lower errors. With more efforts devoted to the long-tail problem, for example, balanced sampling or language adapters, although the initial models still perform worse, the inference phase adjustment can better calibrate the raw classification scores and yield similar performance to the training phase adjustment.

Table 5: Training and inference phase logit adjustments with different A2 models

	en	**fr**	**es**	**it**	**ru**	**zh**	**tt**	**nl**	**tr**	**ky**	**sv**	**avg**
MT + mBERT +	20.4	17.9	13.9	13.0	14.0	34.1	12.4	18.9	14.0	14.3	26.2	18.1
Adjust-Inference	20.1	17.2	13.4	12.2	13.2	34.0	11.5	18.3	13.4	**13.2**	26.1	17.6
Adjust-Train	**19.4**	**16.5**	**12.1**	**11.7**	**12.2**	**31.1**	**11.0**	**17.5**	**12.7**	**13.2**	**24.6**	**16.5**
BS + mBERT +	25.0	20.0	15.4	12.6	13.2	32.9	11.1	17.3	12.6	12.8	22.8	17.8
Adjust-Inference	24.1	19.3	13.5	12.0	12.1	32.3	10.6	**16.7**	12.2	**12.6**	22.2	17.0
Adjust-Train	**23.8**	**19.1**	**13.4**	**11.9**	**11.8**	**32.4**	**10.5**	**16.7**	**12.1**	**12.6**	**21.5**	**16.9**
BS + mBERT + Dual-Adapters +	23.4	18.6	13.4	11.8	11.7	30.8	10.8	16.2	11.6	12.4	21.5	16.5
Adjust-Inference	22.6	17.9	12.7	**11.2**	11.2	**30.1**	10.2	**15.8**	11.4	12.2	**21.1**	**16.0**
Adjust-Train	**22.0**	**17.7**	**12.5**	11.3	**11.1**	30.4	**10.0**	15.9	**11.3**	**12.1**	21.3	**16.0**

4 Related Work

Long-Tail Conventional approaches for addressing the long-tail problem are focused on data sampling methods (Kubat & Matwin, 1997; Chawla et al., 2002; Wallace et al., 2011). On the other hand, normalization approaches for balancing loss have been extensively studied (Morik et al., 1999; Menon et al., 2020). Recently, the long-tail distribution issue has received interest in the context of neural network models, and several approaches are proposed, such as weight normalization (Kang et al., 2019), adaptive margin (Cao et al., 2019), and equalized loss (Tan et al., 2020).

Adapters Adapters were first proposed to learn domain-specific representations in computer vision in a parameter-efficient way (Rebuffi et al., 2017). It was subsequently adopted for NLP tasks to avoid fine-tuning a new model for each task by training an adapter module for each task while sharing the pre-trained language model parameters (Houlsby et al., 2019; Lin et al., 2020). The invertible adapters were proposed in Pfeiffer et al. (2020a;b) to effectively adapt an existing pre-trained multilingual model to unseen languages for multi-task cross-lingual transfer.

Multilingual ASR E2E architectures like LAS (Toshniwal et al., 2018) and Recurrent Neural Transducer (Kannan et al., 2019) have been used for building a multilingual ASR system for a group of Indian languages. In Kannan et al. (2019), language adapters are used to tackle the data imbalance problem, although the improvement of using language adapters is marginal compared to the language vector augmentation. Acoustic vector quantization is also used in a recent work Conneau et al. (2020) for multilingual ASR. A massive multilingual ASR study with more than 50 languages and more than 16,000 hours of speech is presented in Pratap et al. (2020). The two main techniques are data re-sampling and language clusters, which bear some similarities with our balanced sampling and language adapters. Unfortunately, their datasets are not publicly available.

5 Conclusion

In this paper, we introduce the Adapt-and-Adjust (A2), an end-to-end transformer-based framework to address the crucial challenge of long-tail data distribution issues in multilingual speech recognition. A2 consists of three major components, namely language adapters, class imbalance adjustments, and a pretrained multilingual language model. Extensive experiments on the CommonVoice corpus show that A2 significantly outperforms conventional approaches for end-to-end multilingual speech recognition due to 1) better acoustic modeling with adapters and class imbalance adjustments; 2) better language modeling with pretrained multilingual BERT.

REFERENCES

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. Common voice: A massively-multilingual speech corpus. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pp. 4218–4222, 2020.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *CoRR*, abs/1506.03099, 2015.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems*, pp. 1567–1578, 2019.

William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals. Listen, attend and spell. *CoRR*, abs/1508.01211, 2015. URL `http://arxiv.org/abs/1508.01211`.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

Guillem Collell, Drazen Prelec, and Kaustubh Patil. Reviving threshold-moving: a simple plug-in bagging ensemble for binary and multiclass imbalanced data. *arXiv preprint arXiv:1606.08698*, 2016.

Alexis Conneau, Alexei Baevski, Ronan Collobert, Abdelrahman Mohamed, and Michael Auli. Unsupervised cross-lingual representation learning for speech recognition. *arXiv preprint arXiv:2006.13979*, 2020.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9268–9277, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina 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)*, pp. 4171–4186, 2019.

L. Dong, S. Xu, and B. Xu. Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5884–5888, 2018.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pp. 369–376, 2006.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey E. Hinton. Speech recognition with deep recurrent neural networks. *CoRR*, abs/1303.5778, 2013. URL `http://arxiv.org/abs/1303.5778`.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pp. 2790–2799, 2019.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations*, 2019.

Anjuli Kannan, Arindrima Datta, Tara N Sainath, Eugene Weinstein, Bhuvana Ramabhadran, Yonghui Wu, Ankur Bapna, Zhifeng Chen, and Seungji Lee. Large-scale multilingual speech recognition with a streaming end-to-end model. *Proc. Interspeech 2019*, pp. 2130–2134, 2019.

Shigeki Karita, Nelson Enrique Yalta Soplin, Shinji Watanabe, Marc Delcroix, Atsunori Ogawa, and Tomohiro Nakatani. Improving transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration. In *Proc. Interspeech 2019*, pp. 1408–1412, 2019a. doi: 10.21437/Interspeech.2019-1938. URL http://dx.doi.org/10.21437/Interspeech.2019-1938.

Shigeki Karita, Nelson Enrique Yalta Soplin, Shinji Watanabe, Marc Delcroix, Atsunori Ogawa, and Tomohiro Nakatani. Improving transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration. *Proc. Interspeech 2019*, pp. 1408–1412, 2019b.

Suyoun Kim, Takaaki Hori, and Shinji Watanabe. Joint ctc-attention based end-to-end speech recognition using multi-task learning. *CoRR*, abs/1609.06773, 2016. URL http://arxiv.org/abs/1609.06773.

Miroslav Kubat and Stan Matwin. Addressing the curse of imbalanced training sets: One-sided selection. In *ICML*, 1997.

Taku Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 66–75, 2018.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Zhaojiang Lin, Andrea Madotto, and Pascale Fung. Exploring versatile generative language model via parameter-efficient transfer learning. *arXiv preprint arXiv:2004.03829*, 2020.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314*, 2020.

Katharina Morik, Peter Brockhausen, and Thorsten Joachims. Combining statistical learning with a knowledge-based approach-a case study in intensive care monitoring. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pp. 268–277. Morgan Kaufmann Publishers Inc., 1999.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? In *Advances in Neural Information Processing Systems*, pp. 4694–4703, 2019.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. Adapterhub: A framework for adapting transformers. *arXiv preprint arXiv:2007.07779*, 2020a.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. Mad-x: An adapter-based framework for multi-task cross-lingual transfer. *arXiv preprint arXiv:2005.00052*, 2020b.

Vineel Pratap, Anuroop Sriram, Paden Tomasello, Awni Hannun, Vitaliy Liptchinsky, Gabriel Synnaeve, and Ronan Collobert. Massively multilingual asr: 50 languages, 1 model, 1 billion parameters, 2020.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems*, pp. 506–516, 2017.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equalization loss for long-tailed object recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11662–11671, 2020.

Shubham Toshniwal, Tara N. Sainath, Ron J. Weiss, Bo Li, Pedro Moreno, Eugene Weinstein, and Kanishka Rao. Multilingual speech recognition with a single end-to-end model. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr 2018. doi: 10.1109/icassp.2018.8461972. URL http://dx.doi.org/10.1109/ICASSP.2018.8461972.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Byron C Wallace, Kevin Small, Carla E Brodley, and Thomas A Trikalinos. Class imbalance, redux. In *2011 IEEE 11th international conference on data mining*, pp. 754–763. IEEE, 2011.

Guangsen Wang and Khe Chai Sim. Regression-based context-dependent modeling of deep neural networks for speech recognition. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 22(11): 1660–1669, November 2014. ISSN 2329-9290.

Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, et al. Espnet: End-to-end speech processing toolkit. *arXiv preprint arXiv:1804.00015*, 2018.

Zhi-Hua Zhou and Xu-Ying Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Trans. on Knowl. and Data Eng.*, 18(1):63–77, January 2006. ISSN 1041-4347.

A A2 ALGORITHM

The full algorithm of A2 framework with Adjust-Train training phase is outlined in Algorithm 1.

Algorithm 1 Adapt-and-Adjust (A2) with Adjust-Train

Require: $p(\mathcal{T})$: distribution of tasks
Require: θ: an encoder-decoder model
Require: α, λ: step size hyperparameters

randomly initialize θ
transfer language model parameters to decoder parameters in θ
compute class priors π from $p(\mathcal{T})$
while not done **do**
    Sample batch of tasks $\mathbf{x} \sim p(\mathcal{T})$
    Generate language adapter mask $\mathbf{m}$ using the language information from $\mathbf{x}$
    Compute encoder hidden states $\mathbf{h}_{enc}$ using $\mathbf{x}$ and $\mathbf{m}$ by forward propagation on the encoder
    Compute attention logits $\mathbf{y}$ using $\mathbf{h}_{enc}$ and $\mathbf{m}$ by forward propagation on the decoder
    Adjust logit generated from decoder in Equation (5)
    Compute CTC posteriors $P_{\text{CTC}}(\mathbf{y}|\mathbf{h}_{enc})$
    Compute attention loss $\mathcal{L}_{\text{ATTN}}$ in Equation (1)
    Compute multi-task loss $\mathcal{L}_{\text{MTL}}$ using $P_{\text{CTC}}(\mathbf{y}|\mathbf{h}_{enc})$, $\mathcal{L}_{\text{ATTN}}$, and λ in Equation (2)
    Update model $\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{\text{MTL}}$
end while

B ASR MODEL STRUCTURE

We present description of our transformer-based speech recognition model.

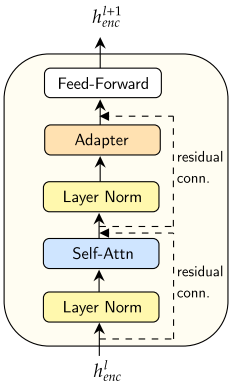

Figure 3: Encoder structure.

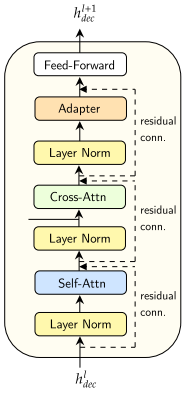

Figure 4: Decoder structure.

Encoder Layer Figure 3 shows the structure of the encoder layer. We use a transformer encoder layer and add an adapter layer on top of the layer norm and self-attention module. We apply two residual connections after the self-attention layer and the adapter layer.

$$o = \text{SelfAttn}(\text{LayerNorm}(\mathbf{h}_{enc}^{l})) + \mathbf{h}_{enc}^{l}, \tag{12}$$

$$\mathbf{h}_{enc}^{l+1} = \text{FeedForward}(\text{Adapter}(\text{LayerNorm}(o)) + o), \tag{13}$$

where $\mathbf{h}_e^l$ is the encoder hidden states of layer l and $\mathbf{h}_e^{l+1}$ is the output of the encoder layer.

Decoder Layer Figure 4 shows the structure of decoder layer. We place the adapter layer on top of the cross-attention model.

$$o_1 = \text{SelfAttn}(\text{LayerNorm}(\mathbf{h}^l_{dec})) + \mathbf{h}^l_{dec}, \tag{14}$$

$$o_2 = \text{CrossAttn}(\mathbf{h}_{enc}, \text{LayerNorm}(o_1)) + o_1 \tag{15}$$

$$\mathbf{h}^{l+1}_{dec} = \text{FeedForward}(\text{Adapter}(\text{LayerNorm}(o_2)) + o_2), \tag{16}$$

where $\mathbf{h}^l_{dec}$ is the decoder hidden states of layer l, $\mathbf{h}^{l+1}_{dec}$ is the output the layer, and $\mathbf{h}_{enc}$ is the encoder hidden states.

C Dataset Distribution

Table 6 shows the data split used for training, validation, and testing. Table 7 shows the frequencies of the labels in the original data and our estimates on the balance sampling. There are only a few data with an occurrence of more than 50k (around 1%). The class distribution on the balance sampling is shifted to the 10k-50k range; thus, there are more labels in the middle of the distribution.

Table 6: Dataset split information

	en	**fr**	**es**	**it**	**ru**	**zh**	**tt**	**nl**	**tr**	**ky**	**sv**
Train (hours)	80.0	50.0	40.0	20.0	15.0	15.0	15.0	10.0	10.0	9.0	4.0
Valid (hours)	109.0	34.0	16.5	8.3	7.3	4.2	2.5	2.5	1.3	1.1	0.5
Test (hours)	10.0	10.0	10.0	8.3	7.3	4.3	2.5	2.5	1.3	1.1	0.5

Table 7: Class frequency in the original data and after applying balanced sampling

	>100k	50k-100k	10k-50k	1k-10k	<1k
Original data	6	9	152	1164	3905
Balance sampling	12	30	306	960	3243

Figure 5 and Figure 6 visualizes the long-tail distribution in more detailed. To better visualize them, we ignore the classes fewer than 5k.

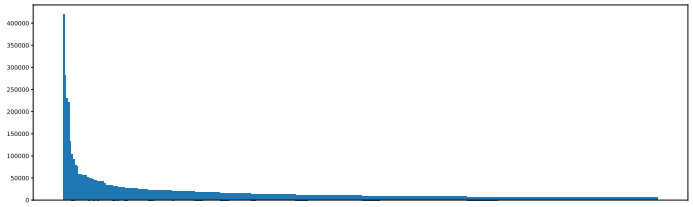

Figure 5: Class Frequency. We only show labels more than 5k

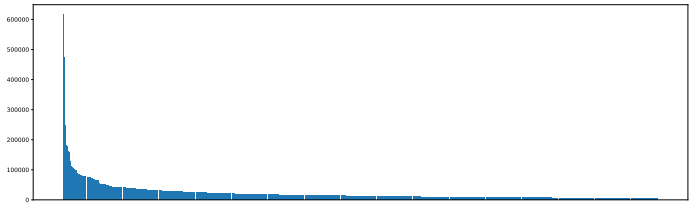

Figure 6: Class Frequency in Balance Sampling. We only show labels more than 5k

D TRANSFER LEARNING USING MBERT

Figure 7 presents the illustration how to modify mBERT model into an autoregressive decoder. We copy embeddings and self-attention parameters of mBERT into a random initialized decoder model.

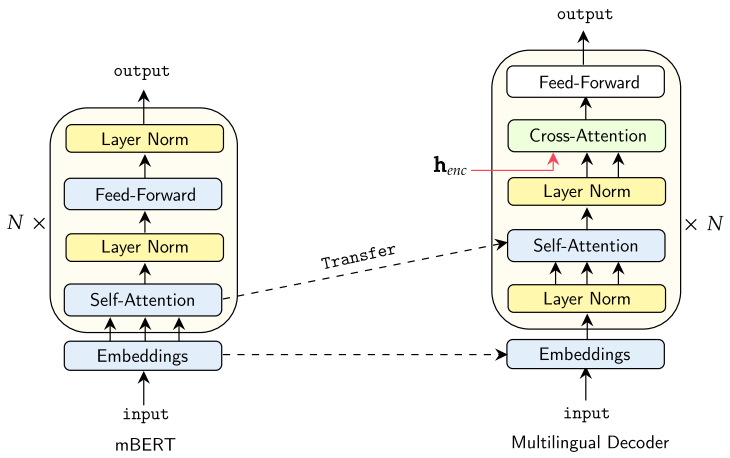

Figure 7: Parameter transfer from a pre-trained multilingual language model to a speech recognition decoder. The dotted line shows the transfer direction from a specific module of the model. h'_{enc} is the encoder hidden states from the last layer of the speech recognition encoder. We omitted the residual connections for simplicity

Table 8: Vocabulary Coverage

	en	**es**	**fr**	**it**	**ky**	**nl**	**ru**	**sv**	**tr**	**tt**	**zh**	**multi**
Coverage (%)	74.4	70.8	73.6	73.9	91.1	64.7	90.2	53.8	57.1	91.3	97.7	54.9
# Match	181	192	281	164	154	110	156	86	93	157	2,330	2,875
# Vocab	243	271	382	222	169	170	173	160	163	172	2,386	5,237

E LOSS DYNAMICS

We plot the validation losses from A2 and other baselines in Figures 8 and 9.

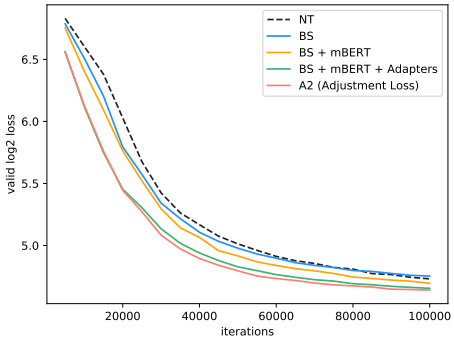

Figure 8: Validation losses

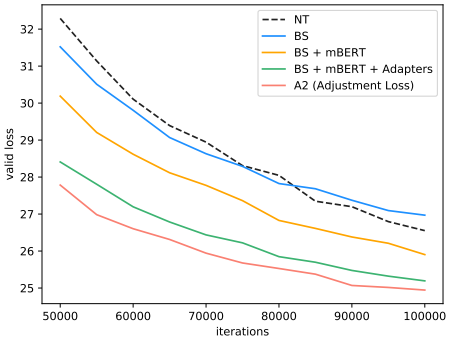

Figure 9: Closer look on Validation losses

F FEW-SHOT SETTING

We also investigate the effectiveness of the A2 framework in the few-shot setting. We take five-hour speech data on each language. The test results are shown in Table 9. Using mBERT as the speech decoder consistently improves the performance of the low-resource monolingual scenario. Interestingly in Chinese, without using mBERT as a decoder, the model seems not to be able to learn useful information. The mBERT decoder yields an absolute CER reduction of 17.2%.

Further performance gain can be achieved with mBERT as the decoder, indicating the multilingual language model is complementary to the shared acoustic knowledge learned from the multilingual speech data. Our proposed A2 framework with language adapters and adaptive loss leads to consistent performance gain across all languages. The average performance translates to a significant CER reduction of 4% compared to the baseline. We can further improve the performance by applying the Adjust-Inference.

Table 9: Character Error Rate (CER) on five hour CommonVoice Training data

	en	**fr**	**es**	**it**	**ky**	**nl**	**ru**	**sv**	**tr**	**tt**	**zh**	**avg**
Monolingual												
Standard Multilingual Traning (MT)	69.6	61.1	56.5	49.8	43.7	57.5	49.6	56.1	49.6	32.8	96.4	56.5
MT + mBERT	67.3	56.1	51.0	45.4	39.6	52.9	42.8	55.1	46.1	29.6	73.2	50.8
Multilingual												
MT	41.4	33.5	25.2	22.3	18.9	24.9	21.9	28.9	18.7	15.5	43.0	26.7
MT + mBERT	40.9	32.6	24.3	21.0	17.4	23.8	19.9	26.7	17.3	14.5	42.1	25.5
MT + mBERT + Adapters	36.7	28.6	21.3	18.3	15.1	20.9	17.2	23.9	15.1	12.9	38.2	22.6
A2 (Adjust-Inference)	**36.5**	**28.4**	**21.0**	**18.0**	**15.0**	**20.6**	**16.9**	**23.7**	**14.8**	**12.7**	**38.0**	**22.3**

G HYPER-PARAMETER SETTING

The hyper-parameters used in the training is The hyper-parameters used in training are shown in Table 10.

Table 10: Hyper-parameter Setting for Training

hyper-parameters	**value**
encoder-layer	6
encoder-units	2048
decoder-layers	6
decoder-units	3072
encoder-attention-dim	256
decoder-attention-dim	756
attention-heads	4
label-smoothing	0.1
batch-size	32 (for random sampling) 6 (for balanced sampling)
maximum-length-in	512
maximum-length-out	150
optimizer	noam
warmup-step	25000
accum-grad	2 (for random sampling) 11 (for balanced sampling)
grad-clip	5