



# Text-only Domain Adaptation using Unified Speech-Text Representation in Transducer

Lu Huang\*, Boyu Li\*, Jun Zhang, Lu Lu, Zejun Ma

ByteDance

{huanglu.thu19, liboyu.622}@bytedance.com

## Abstract

Domain adaptation using text-only corpus is challenging in end-to-end(E2E) speech recognition. Adaptation by synthesizing audio from text through TTS is resource-consuming. We present a method to learn Unified Speech-Text Representation in Conformer Transducer(USTR-CT) to enable fast domain adaptation using the text-only corpus. Different from the previous textogram method, an extra text encoder is introduced in our work to learn text representation and is removed during inference, so there is no modification for online deployment. To improve the efficiency of adaptation, single-step and multi-step adaptations are also explored. The experiments on adapting LibriSpeech to SPGISpeech show the proposed method reduces the word error rate(WER) by relatively 44% on the target domain, which is better than those of TTS method and textogram method. Also, it is shown the proposed method can be combined with internal language model estimation(ILME) to further improve the performance.

**Index Terms:** automatic speech recognition, text-only, domain adaptation, conformer transducer

## 1. Introduction

In recent years, E2E models have achieved significant improvements in automatic speech recognition(ASR)[1, 2, 3]. Compared with hybrid models, where acoustic, pronunciation, and language models(LMs) are built and optimized separately, E2E models have achieved promising performance by directly mapping speech features into word sequences. There are some popular E2E models, including connectionist temporal classification[4, 5], recurrent neural network transducer(RNN-T)[2, 6, 7], and attention-based encoder-decoder[8, 9, 10].

However, unlike hybrid models which can adapt to new domains by training LMs using text-only data, E2E model has difficulty in domain adaptation using text-only data. Besides, the E2E models are trained with paired speech-text data, so their generalization ability to different contents is limited, and the performance degrades when a mismatch exists between source and target domains. To overcome this, the most promising approach is to adapt the E2E model using text-only data, because it is much easier to collect text-only data than paired speech-text data in the target domain.

Several methods have been proposed to adapt the E2E model to new domain. The most common solution is to train an external LM using text corpus in the target domain and integrate it into the E2E model during inference, such as shallow fusion[11], density ratio fusion[12], deep fusion[13], cold fusion[14], and internal LM(ILM) estimation based fusion[15,

16]. Nevertheless, all these methods involve an external LM during inference, and the computation cost of decoding is increased.

Other methods attempt to directly update the E2E model to avoid changing the decoding. Synthesizing paired speech-text data using TTS is a common solution[17, 18], but the process is complex, also the storage and computation cost increases. Recently, a text-to-mel-spectrogram generator is proposed in E2E model[19] for replacing TTS. The method proposed in [20] inserts a temporary LM layer into the prediction network and the LM loss is used for adapting with text-only data. Internal LM adaptation(ILMA)[21] is proposed to fine-tune the parameters of internal LM with an additional LM loss.

Alternative approaches focus on creating a shared embedding space by joint training for two modalities, i.e., speech and text, and have been reported to improve ASR performance without increasing model parameters or decoding complexity[22, 23]. Recent works[24, 25] involve this idea to make a consistent representation between text and speech features in text-only adaptation tasks.

Inspired by [22, 24, 25], we proposed a method to learn the unified speech-text representation in Conformer Transducer(USTR-CT) for fast text-only domain adaptation. Separated encoders are adopted to learn a shared representation for speech and text features respectively, and a shared encoder is used for the fusion of speech and text representation. At the same time, different representation units are explored and phoneme representation performs best in the target domain. To improve the efficiency of adaptation, single-step and multi-step adaptations are also explored. Finally, We observe 44% relative WER reduction in the target domain with unpaired text data. In addition, combined with ILME, the proposed method can obtain further gains.

The paper is organized as follows: Section 2 gives a brief introduction about related work. The proposed USTR-CT is discussed in Section 3, followed by experiments and discussions in Section 4.

## 2. Related work

### 2.1. Speech-text joint training for ASR

Several methods have been proposed to train E2E model with speech and text modalities[22, 23, 26, 27, 28, 29, 30, 31].

The recent JOIST[23] explores joint training with a combination of losses computed on the supervised paired speech-text data and the unpaired text data in cascaded encoder based streaming ASR framework[32]. The input unpaired text representation are up-sampled with a simple and parameter-free duration model, and then fed to a text encoder after masking. The output of the text encoder can be fed to the first-pass decoder,

\* Equal contribution.

or to the shared encoder and second-pass decoder. However, the experiments were conducted on a multi-domain corpus and the performance of long-tail rare words are evaluated. In this work, we mainly focus on domain adaptation, i.e., transferring the model to the target domain with text-only data.

In order to ensure that the representations learned from speech and text features are aligned, MAESTRO[22] introduces a consistency loss to align two types of representations using the paired data. The method has achieved significant improvements on ASR and speech translation tasks. Different from our work, it focuses on the self-supervised training to obtain a better pre-trained model rather than domain adaptation. Besides, the additional duration model increases the complexity of training.

## 2.2. Text-only domain adaptation

In this subsection, we give a brief introduction to text-only domain adaptation methods without using external LM.

**TTS adaptation** generates paired speech-text data from the target domain text for fine-tuning[17, 18]. However, the number of speakers is limited for TTS and training a reliable multi-speaker TTS model is time-consuming and computationally expensive. Besides, saving the synthesized data also increases storage costs. Moreover, due to the mismatch between synthetic and real audio, additional issues may be also introduced. To mitigate the above problems, recently a text-to-spectrogram front-end composed of a text-to-mel-spectrogram generator was introduced in ASR model[19]. In this way, on-the-fly generating spectrograms from text-only data is possible, and the mismatch between real and synthetic audio is not obvious as before. However, it still needs to pay attention to the quality of the spectrogram enhancer in the generator during training.

**ILMA** propose to adapt the E2E model[21] by fine-tuning the ILM with text-only data, and parameter regularization is added to avoid over-fitting. Also, it is essential to perform ILM training(ILMT)[33] in addition to ASR loss before ILMA to ensure that the ILM behaves like a standalone LM. As only the last linear layer of the jointer network is updated, ILMA’s performance on the target domain is also limited.

**Textogram** was proposed in [24], where the text representation is created by repeating one-hot embedding of text tokens by a fixed number of times. The textogram features are stacked together with standard speech features. When training using text-only data, the speech features are set to zero. And when training using paired speech-text data, the textogram features are set to zero. Due to the concatenation of speech features and textogram, it is necessary to concatenate zero textogram features with speech features during inference. However, in our work, with separated encoders, the input is either text features or speech features during training, and inference can be performed directly without any modifications. Besides, updating only the jointer performs best in textogram, while the jointer, predictor, and even encoder can be adapted to the target domain with better performance. In addition to grapheme representation, subword and phoneme representations are also explored in our work.

## 3. Training and adapting methods

### 3.1. Model architecture

For the standard RNN-T, speech-text pairs are used to train the model. Let  $\mathbf{x}_{1:T}^{\text{speech}}$  be the audio features like Fbank, and  $\mathbf{y}_{0:u-1}$  be the previous tokens, the output of RNN-T at frame  $t$  and step

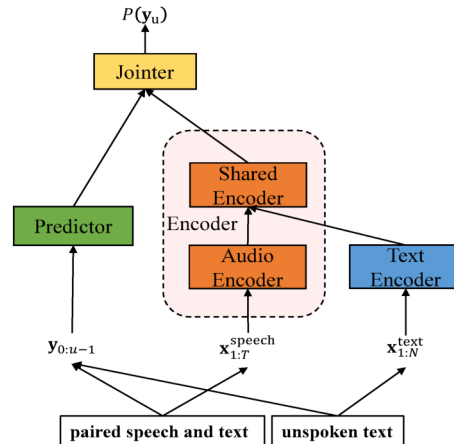


Figure 1: The model structures of USTR-RNN-T.

$u$  is computed by

$$\mathbf{h}_{1:T}^{\text{enc}} = \text{Encoder}(\mathbf{x}_{1:T}^{\text{speech}}), \quad (1)$$

$$\mathbf{h}_u^{\text{pred}} = \text{Predictor}(\mathbf{y}_{0:u-1}), \quad (2)$$

$$\mathbf{h}_{t,u}^{\text{joint}} = \text{Jointer}(\mathbf{h}_t^{\text{enc}}, \mathbf{h}_u^{\text{pred}}). \quad (3)$$

Then with a Softmax layer on the  $\mathbf{h}_{t,u}^{\text{jointer}}$  and forward-backward algorithm[7], Transducer loss, which is the sum probability  $P(\mathbf{y}|\mathbf{x})$  of all possible alignment paths  $\pi$ , is computed as the training objective function

$$\mathcal{L}_{\text{rnn-t}} = -\log \sum_{\pi \in \Pi(\mathbf{y})} P(\pi|\mathbf{x}_{1:T}^{\text{speech}}), \quad (4)$$

where  $\mathbf{y} = (y_1, \dots, y_U)$  is the label sequence,  $U$  is the number of target tokens, and  $\Pi(\mathbf{y})$  is the alignment path sets.

To involve the text-only corpus during training, the RNN-T encoder is split into two parts, named `AudioEncoder` and `SharedEncoder`, and an extra `TextEncoder` is introduced to model text features  $\mathbf{x}_{1:N}^{\text{text}}$ , which is illustrated in Figure 1 as *USTR-RNN-T*. The Transducer loss can be computed the same way as paired speech-text corpus,

$$\mathbf{h}_{1:N}^{\text{enc, text}} = \text{SharedEncoder}(\text{TextEncoder}(\mathbf{x}_{1:N}^{\text{text}})), \quad (5)$$

$$\mathbf{h}_{n,u}^{\text{joint, text}} = \text{Jointer}(\mathbf{h}_n^{\text{enc, text}}, \mathbf{h}_u^{\text{pred}}), \quad (6)$$

where  $\mathbf{x}_{1:N}^{\text{text}}$  can be grapheme/sub-word/phoneme representations. As the extra `TextEncoder` can be removed during inference, the proposed method doesn’t need any modification for online deployment.

In our experiments, `SharedEncoder` consists twelve non-streaming Conformer[34] layers, so the baseline is noted as `Conformer Transducer(CT)`. `AudioEncoder` has two `Conv2d` layers with stride of 2 and a linear projection layer, resulting a time reduction of 4. `TextEncoder` contains an embedding layer and a Transformer layer. For CT’s prediction network, 2-layer LSTM is adopted, and RNN-T’s jointer network is a feed-forward layer.

### 3.2. Training

To enforce speech-text modality matching in a joint embedding space for `SharedEncoder`, paired speech-text samples are needed for training `AudioEncoder` and `TextEncoder`.

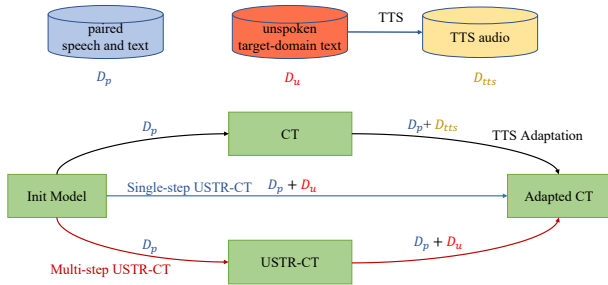


Figure 2: The adaptation processes of TTS, multi-step and single-step USTR-CT.

When `TextEncoder` is introduced in the training, the paired speech-text in the training corpus is used as unspoken text with a random probability  $p$  by using the text features instead of the audio features.

Three types of text features are considered in this work. The first one is grapheme features, which is similar as the `textogram` in [24]. The second one is subword features, which is the same as the output vocabulary of CT, and is generated using `subword-nmt`[35]<sup>1</sup>. The final one is phoneme features, which is generated by a Grapheme-to-Phoneme system. For English in this work, `g2pE2` is used.

To simulate the duration of speech features, the text features are repeated a fixed number of times, which is the same as that in [24, 23]. Also, text features are masked to prevent the `TextEncoder` from memorizing the grapheme/subword sequence blindly [24, 23]. However, the masking method differs from that in [24, 23] by applying on repeated text features. It is found that masking before repeating brings better performance.

During training, a mini-batch containing both text and speech features is fed into the model. Besides, ILMT loss is chosen as an optional auxiliary loss, and the overall loss is

$$\mathcal{L} = \mathcal{L}_{\text{rnn-t}} + \lambda \mathcal{L}_{\text{ilmt}}, \quad (7)$$

where  $\lambda$  is the weight corresponding to ILMT loss, which is set to 0.2 in all experiments.

### 3.3. Adapting

When the text-only corpus is used for adapting the CT model to a new domain, two adaptation strategies are investigated in this work, as illustrated in Figure 2.

#### 3.3.1. Multi-step adaptation

As illustrated in the bottom part of Figure 2, multi-step adaptation using USTR-CT contains two steps.

In the first step, paired speech-text data is used to train a USTR-CT, where each sample is fed into the `TextEncoder` by using text features instead of audio features with a random probability  $p$  to train the `TextEncoder`. The probability  $p$  is set to 0.15 in the experiments.

In the second step, i.e., the stage of adapting, paired speech-text data and unspoken text are both used in each mini-batch with a ratio of 1:1. The ratio can be further tuned to obtain better performance on the target domain, which is left for future discussion. The paired speech-text data is used to maintain the

<sup>1</sup><https://github.com/rsennrich/subword-nmt>

<sup>2</sup><https://github.com/Kyubyong/g2p>

performance on source domain. In this step, the parameters of `AudioEncoder` and `SharedEncoder` (i.e., the `Encoder` of CT) are kept constant, while `Joiner` and `Predictor` are trained to adapt to new domain. Due to the existence of USTR-CT model after first step, it is more convenient for adapting to other domains when there is multi-domain scenario.

#### 3.3.2. Single-step adaptation

As shown in the middle part of Figure 2, single-step USTR-CT trains an adapted CT model from random initialization directly. Similar to multi-step USTR-CT, paired speech-text data is also fed into the `TextEncoder` by a probability  $p = 0.15$ . Also, the ratio between paired speech-text data and unspoken text is still 1:1 to be consistent with multi-step adaptation.

## 4. Experiments and results

### 4.1. Experimental setup

The experiments are conducted on LibriSpeech[36] and SPGISpeech[37] corpora. SPGISpeech contains 5,000 hours of financial audio. In this work, only the transcribed text of SPGISpeech is used for text-only domain adaptation, which has 1.7M utterances. Two versions of the text are created in the experiments, noted as `Large(L)` and `Small(S)`, where the former contains the full 1.7M utterances and the latter contains a subset of 280k utterances, which is almost the same as the number of LibriSpeech utterances. Besides, the TTS audios are synthesized from the `Small` subset using an in-house engine to indicate that TTS is resource-consuming.

For the audio features, the 80-dim filter-bank(Fbank) is used and `Spec-Augment`[38] is applied on Fbank features. The text features are masked with a probability of 0.15 before repeating. The model’s structure is described as that in Section 3, and the output of RNN-T is 4,048 subword units. All models are trained with PyTorch[39]. WER is evaluated on LibriSpeech `test-clean/test-other` sets and SPGISpeech `val` set to measure the ASR performance on source and target domain.

### 4.2. Baseline systems

A CT model is trained on LibriSpeech, which achieves a WER of 23.55% on SPGISpeech `val` set. With TTS based adaptation, as shown in the top part of Figure 2, the WER is reduced to 14.99% by 36.35% relatively. Besides, the `textogram`[24] method is also evaluated in this work, which achieves a WER of 23.94%. `Textogram` based adaptation, where the encoder and joiner are kept constant, is trained with text-only corpus. And after adaptation, the WER is reduced by relatively 33.25%/37.80% when using the S/L subset respectively.

The proposed multi-step USTR-CT is firstly trained with grapheme representation by masking with a rate of 0.15 and repeating four times. As illustrated in Table 1, the proposed USTR-CT achieves a WER of 22.72% on SPGISpeech `val` set before adaptation, which is better than the `textogram`. After adaptation, the proposed method not only performs better on the target domain, but also achieves the best performance on LibriSpeech test sets, as the paired speech-text data is used to maintain the performance of source domain during adaptation. The results indicate that the extra text encoder in USTR-CT and joiner adaptation in the second step are beneficial. Also, the proposed method outperforms TTS based adaptation on both source and target domains, with relative WER reductions of 2.54%~5.45%.

Table 1: The WER(%) of different systems on LibriSpeech test sets and SPGISpeech val set. Text adaptation L/S corresponds to the Large/Small subset of SPGISpeech’s transcribed text.

model	LibriSpeech test clean/other	SPGISpeech val
CT	3.99/8.28	23.55
+ TTS adaptation	3.85/8.12	14.99
textogram baseline [24]	4.18/8.84	23.94
+ text adaptation(S)	5.12/10.10	15.98
+ text adaptation(L)	4.43/8.98	14.89
multi-step USTR-CT	3.76/8.15	22.72
+ text adaptation(S)	3.66/8.00	14.89
+ text adaptation(L)	<b>3.64/7.84</b>	<b>14.61</b>

### 4.3. Representation units

Table 2: The WER(%) of multi-step USTR-CT using different representation units for text features.

model	LibriSpeech test clean/other	SPGISpeech val
grapheme repeat 4	3.76/8.15	22.72
+ text adaptation(L)	<b>3.64/7.84</b>	14.61
phoneme repeat 3	4.10/8.32	23.11
+ text adaptation(L)	3.80/7.96	13.41
phoneme repeat 4	3.82/8.18	22.30
+ text adaptation(L)	<b>3.64/7.80</b>	<b>13.38</b>
phoneme repeat 5	3.82/8.08	22.17
+ text adaptation(L)	3.71/7.82	13.42
subword repeat 4	4.20/8.49	23.92
+ text adaptation(L)	3.80/8.27	15.00

Different representation units are explored for multi-step USTR-CT, where the repeating number of phoneme representation is also investigated. As illustrated in Table 2, for the repeating number of 4, phoneme representation performs best on target domain(WER 13.38% vs. 14.61%/15.00%). This may be due to the phoneme representation is more relevant to Fbank features, and thus the learning of unified speech-text representation can be more easier. Besides, we changed the repeating number from 4 to 3/5 and no further gains were observed.

### 4.4. Multi-step vs. single-step

We compared single-step USTR-CT with multi-step USTR-CT and TTS based adaptation, and the results are illustrated in Table 3. It is shown that single-step USTR-CT performs best on both source and target domain. As the shared encoder is also adapted to target domain, the single-step USTR-CT performs even better than multi-step USTR-CT. Besides, the extra text of SPGISpeech also benefits the source domain.

### 4.5. Combination with ILME

As the text of target domain is already involved in training of the CT model, the benefit of external LM may be discounted. We have trained an LSTM LM and evaluated the performance

Table 3: The WER(%) of multi-/single-step USTR-CT.

model	LibriSpeech test clean/other	SPGISpeech val
CT + TTS adaptation	3.85/8.12	14.99
multi-step USTR-CT	3.82/8.18	22.30
+ text adaptation(L)	3.64/7.80	13.38
single-step USTR-CT(L)	<b>3.07/7.13</b>	<b>13.25</b>

Table 4: The WER(%) of different models with ILME.

model	SPGISpeech val
CT	23.55
+ ILME	13.83
CT + TTS adaptation	14.99
+ ILME	11.34
multi-step USTR-CT	13.38
+ ILME	<b>10.05</b>
single-step USTR-CT	13.25
+ ILME	10.80

of ILME using different CT models. As illustrated in Table 4, ILME brings WER reductions of 41.27%/24.35% on baseline CT and TTS based models. For multi-/single-step USTR-CT, ILME also reduces the WER by 24.89%/18.49% respectively. This indicated that the proposed USTR is able to combine with ILME to further improve the performance on the target domain.

It is noticed that multi-step USTR-CT performs better than single-step USTR-CT when combined with ILME, which is different from the results without ILME in Section 4.4. We assume that the ILM score of single-step USTR-CT is not accurate, as the encoder also captures the linguistic information of target domain during training. Besides, as the encoder is frozen during the 2nd step, multi-step USTR-CT is more suitable for training with a large scale text corpus.

## 5. Conclusions

In this work, an extra text encoder is introduced for text-only domain adaptation, which outperforms the TTS adaptation by 11.61% relatively when using phoneme representation. Compared to TTS adaptation, the proposed USTR-CT is efficient and resource-saving for fast domain adaptation. Besides, USTR-CT is able to adapt to the target domain with a single-step training and combine with ILME to obtain further gains. Although experiments were conducted on non-streaming models in this work, the method is still applicable for streaming ASR. With the separated speech and text encoders, the similarity between the speech and text modalities can be considered to further improve the performance, which is left as a discussion in future work.

## 6. References

- [1] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina *et al.*, “State-of-the-art speech recognition with sequence-to-sequence models,” in *ICASSP*, 2018, pp. 4774–4778.
- [2] T. N. Sainath, Y. He, B. Li, A. Narayanan, R. Pang, A. Bruguier,

- S.-y. Chang, W. Li, R. Alvarez, Z. Chen *et al.*, “A streaming on-device end-to-end model surpassing server-side conventional model quality and latency,” in *ICASSP*, 2020, pp. 6059–6063.
- [3] J. Li *et al.*, “Recent advances in end-to-end automatic speech recognition,” *APSIPA Transactions on Signal and Information Processing*, vol. 11, no. 1, 2022.
- [4] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in *ICML*, 2006, pp. 369–376.
- [5] J. Li, G. Ye, A. Das, R. Zhao, and Y. Gong, “Advancing acoustic-to-word ctc model,” in *ICASSP*, 2018, pp. 5794–5798.
- [6] J. Li, R. Zhao, Z. Meng, Y. Liu, W. Wei, S. Parthasarathy, V. Mazalov, Z. Wang, L. He, S. Zhao *et al.*, “Developing rnn-t models surpassing high-performance hybrid models with customization capability,” in *Interspeech*, 2020, pp. 3590–3594.
- [7] A. Graves, “Sequence transduction with recurrent neural networks,” in *ICML Representation Learning Workshop*, 2012.
- [8] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, “Attention-based models for speech recognition,” in *NeurIPS*, vol. 28, 2015.
- [9] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang *et al.*, “A comparative study on transformer vs rnn in speech applications,” in *ASRU 2019*, 2019, pp. 449–456.
- [10] J. Li, Y. Wu, Y. Gaur, C. Wang, R. Zhao, and S. Liu, “On the comparison of popular end-to-end models for large scale speech recognition,” in *Interspeech*, 2020, pp. 1–5.
- [11] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates *et al.*, “Deep speech: Scaling up end-to-end speech recognition,” *arXiv preprint arXiv:1412.5567*, 2014.
- [12] E. McDermott, H. Sak, and E. Variani, “A density ratio approach to language model fusion in end-to-end automatic speech recognition,” in *ASRU 2019*, 2019, pp. 434–441.
- [13] C. Gulcehre, O. Firat, K. Xu, K. Cho, L. Barrault, H.-C. Lin, F. Bougares, H. Schwenk, and Y. Bengio, “On using monolingual corpora in neural machine translation,” *arXiv preprint arXiv:1503.03535*, 2015.
- [14] A. Sriram, H. Jun, S. Satheesh, and A. Coates, “Cold fusion: Training seq2seq models together with language models,” in *Interspeech*, 2018, pp. 387–391.
- [15] E. Variani, D. Rybach, C. Allauzen, and M. Riley, “Hybrid autoregressive transducer (hat),” in *ICASSP*, 2020, pp. 6139–6143.
- [16] Z. Meng, S. Parthasarathy, E. Sun, Y. Gaur, N. Kanda, L. Lu, X. Chen, R. Zhao, J. Li, and Y. Gong, “Internal language model estimation for domain-adaptive end-to-end speech recognition,” in *SLT*, 2021, pp. 243–250.
- [17] Y. Huang, J. Li, L. He, W. Wei, W. Gale, and Y. Gong, “Rapid rnn-t adaptation using personalized speech synthesis and neural language generator,” in *Interspeech*, 2020, pp. 1256–1260.
- [18] C. Peyser, S. Mavandadi, T. N. Sainath, J. Apfel, R. Pang, and S. Kumar, “Improving tail performance of a deliberation e2e asr model using a large text corpus,” in *Interspeech*, 2020, pp. 4921–4925.
- [19] V. Bataev, R. Korostik, E. Shabalin, V. Lavrukhin, and B. Ginsburg, “Text-only domain adaptation for end-to-end asr using integrated text-to-mel-spectrogram generator,” *arXiv preprint arXiv:2302.14036*, 2023.
- [20] J. Pyllkkönen, A. Ukkonen, J. Kilpikoski, S. Tamminen, and H. Heikinheimo, “Fast Text-Only Domain Adaptation of RNN-Transducer Prediction Network,” in *Interspeech*, 2021, pp. 1882–1886.
- [21] Z. Meng, Y. Gaur, N. Kanda, J. Li, X. Chen, Y. Wu, and Y. Gong, “Internal Language Model Adaptation with Text-Only Data for End-to-End Speech Recognition,” in *Interspeech*, 2022, pp. 2608–2612.
- [22] Z. Chen, Y. Zhang, A. Rosenberg, B. Ramabhadran, P. J. Moreno, A. Bapna, and H. Zen, “MAESTRO: Matched Speech Text Representations through Modality Matching,” in *Interspeech*, 2022, pp. 4093–4097.
- [23] T. N. Sainath, R. Prabhavalkar, A. Bapna, Y. Zhang, Z. Huo, Z. Chen, B. Li, W. Wang, and T. Strohmaier, “Joist: A joint speech and text streaming model for asr,” in *SLT 2022*, 2023, pp. 52–59.
- [24] S. Thomas, B. Kingsbury, G. Saon, and H.-K. J. Kuo, “Integrating text inputs for training and adapting rnn transducer asr models,” in *ICASSP*, 2022, pp. 8127–8131.
- [25] H. Sato, T. Komori, T. Mishima, Y. Kawai, T. Mochizuki, S. Sato, and T. Ogawa, “Text-only domain adaptation based on intermediate etc,” in *Interspeech*, 2022, pp. 2208–2212.
- [26] A. Bapna, Y.-a. Chung, N. Wu, A. Gulati, Y. Jia, J. H. Clark, M. Johnson, J. Riesa, A. Conneau, and Y. Zhang, “Slam: A unified encoder for speech and language modeling via speech-text joint pre-training,” *arXiv preprint arXiv:2110.10329*, 2021.
- [27] A. Bapna, C. Cherry, Y. Zhang, Y. Jia, M. Johnson, Y. Cheng, S. Khanuja, J. Riesa, and A. Conneau, “mslam: Massively multilingual joint pre-training for speech and text,” *arXiv preprint arXiv:2202.01374*, 2022.
- [28] Y. Tang, H. Gong, N. Dong, C. Wang, W.-N. Hsu, J. Gu, A. Baevski, X. Li, A. Mohamed, M. Auli *et al.*, “Unified speech-text pre-training for speech translation and recognition,” in *ACL*, 2022, pp. 1488–1499.
- [29] S. Thomas, H.-K. J. Kuo, B. Kingsbury, and G. Saon, “Towards reducing the need for speech training data to build spoken language understanding systems,” in *ICASSP*, 2022, pp. 7932–7936.
- [30] Y.-A. Chung, C. Zhu, and M. Zeng, “Splat: Speech-language joint pre-training for spoken language understanding,” in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 1897–1907.
- [31] J. Ao, R. Wang, L. Zhou, C. Wang, S. Ren, Y. Wu, S. Liu, T. Ko, Q. Li, Y. Zhang *et al.*, “Speech5: Unified-modal encoder-decoder pre-training for spoken language processing,” in *ACL*, 2022, pp. 5723–5738.
- [32] A. Narayanan, T. N. Sainath, R. Pang, J. Yu, C.-C. Chiu, R. Prabhavalkar, E. Variani, and T. Strohmaier, “Cascaded encoders for unifying streaming and non-streaming asr,” in *ICASSP*, 2021, pp. 5629–5633.
- [33] Z. Meng, N. Kanda, Y. Gaur, S. Parthasarathy, E. Sun, L. Lu, X. Chen, J. Li, and Y. Gong, “Internal language model training for domain-adaptive end-to-end speech recognition,” in *ICASSP*, 2021, pp. 7338–7342.
- [34] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, “Conformer: Convolution-augmented transformer for speech recognition,” in *Interspeech*, 2020.
- [35] R. Sennrich, B. Haddow, and A. Birch, “Neural machine translation of rare words with subword units,” in *ACL*, 2016, pp. 1715–1725.
- [36] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: an asr corpus based on public domain audio books,” in *ICASSP*, 2015, pp. 5206–5210.
- [37] P. K. O’Neill, V. Lavrukhin, S. Majumdar, V. Noroozi, Y. Zhang, O. Kuchaiev, J. Balam, Y. Dovzhenko, K. Freyberg, M. D. Shulman, B. Ginsburg, S. Watanabe, and G. Kucsko, “Spgispeech: 5,000 hours of transcribed financial audio for fully formatted end-to-end speech recognition,” in *Interspeech*, 2021, pp. 1434–1438.
- [38] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, “SpecAugment: A simple data augmentation method for automatic speech recognition,” in *Interspeech*, 2019, pp. 2613–2617.
- [39] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, “Pytorch: An imperative style, high-performance deep learning library,” in *NeurIPS*, vol. 32, 2019.