Effective Training of Attention-based Contextual Biasing Adapters with Synthetic Audio for Personalised ASR

Burin Naowarat[1,2*], *Philip Harding*[2], *Pasquale D'Alterio*[2], *Sibo Tong*[2], *Bashar Awwad Shiekh Hasan*[2]

[1]Chulalongkorn University, Bangkok, Thailand
[2]Amazon Alexa, Cambridge, UK

6270145221@student.chula.ac.th, hasbasha@amazon.com

Abstract

Contextual biasing (CB) is an effective approach for contextualising hidden features of neural transducer ASR models to improve rare word recognition. CB relies on relatively large quantities of relevant human annotated natural speech during training, limiting its effectiveness in low-resource scenarios. In this work, we propose a novel approach that reduces the reliance on real speech by using synthesised audios for training CB adapters. We introduce a projection module (PM) that transforms encoder features of synthesised audios prior to CB training to better match real speech. We penalise PM with consistency regularisation to encourage higher similarity between features of real and synthesised speech. The proposed method maintains the same performance on both named-entity and general datasets while using half of the real speech data for CB training. Furthermore, we show a 16% word error rate reduction when the full real-speech training dataset is extended with synthetic utterances.

Index Terms: speech recognition, contextual biasing, personalised ASR

1. Introduction

Neural Transducer models, including Recurrent Neural Network Transducer (RNN-T) and Conformer-Transducer (C-T), have achieved state-of-the-art accuracy on a range of Automatic Speech Recognition (ASR) tasks and as such have been widely adopted [1, 2]. Neural transducer models are typically trained to directly estimate textual units such as graphemes or sub-words [3]. This simplifies the training process compared to existing 'hybrid' ASR models as lexicons and alignments are no longer required, but reduces the adaptability of the system. Despite strong performance on many tasks, rare word recognition remains an open issue, with neural transducers often failing to recognise words that were seen infrequently during training.

Rare word recognition is critical for a range of tasks, including virtual assistants, where named entities such as contact names, song titles and appliance names are often also rare words. Recognition of rare words can be improved to a certain degree using language models, either to directly bias posteriors in the search (shallow-fusion) or to re-score n-best lists. These approaches have limitations, however; shallow fusion interpolation weights require careful tuning [4], and in the case of rescoring methods the rare words might have already been pruned from the candidate list making correction impossible [5].

Recent work proposed to improve rare word recognition by directly adapting models in the latent space via contextual biasing (CB) [6, 7, 8, 9, 10]. CB steers the predictions of transducer models towards a given list of entities from a given context by first encoding these entities into a model-interpretable format and then performing cross-attention between the encoded text features and encoder outputs of the neural transducer. The encoder outputs are then biased by adding the output of the cross attention. Training CB adapters relies on transcribed speech recordings of utterances containing relevant entities, which are often time consuming and expensive to obtain, while relevant contextual entities can be retrieved from personal catalogues, e.g. playlist names, favourite movies or saved addresses.

On the other hand, the use of synthesised audios for training ASR models has gained attention as Text-to-Speech (TTS) models become capable of synthesising high-fidelity audio [11, 12, 13, 14, 15]. Despite real and synthesised audio being difficult to distinguish for human listeners [16], synthesised audio is still sufficiently different in the model feature space that models trained on such audio often fail to generalise well to natural speech [4]. Recent work looks to solve this problem by limiting the sample size of synthesised audios [17], adding consistency regularisation [18, 19], performing multi-stage training [20, 21], and separating normalisation statistics [16].

In this work, we propose a simple feature transformation to effectively use synthesised audios for training CB adapters. First, we introduce a projection module (PM) to reduce the mismatch between encoded features of real and synthesised speech. During training, the PM is added between the encoder and CB adapter and is jointly trained with the CB adapter while the rest of the network is frozen. As the PM is only applied to TTS speech, it is not required for inference and so is discarded after training. Second, we further penalise dissimilarities between encoder features of real and synthesised speech by imposing consistency regularisation to the PM. We present experimental results for two alternatives: i.) gradient reversal (GR) [22] and ii.) contrastive loss (CL) [23]. GR and CL only affect the PM outputs and have no impact on other components.

Experimental results on in-house test sets show that our proposed PM prevented the models from over-fitting to synthesised audios to the extent that we were able to reduce the amount of human speech required for training by 50%. By extending the full human speech corpus with utterances with synthetic audio and transcripts we were able to increase word error rate reductions on named entities by a further 16%.

2. Contextual Biasing RNN-T

A Contextual Biasing Transducer (CB-T) is as a neural transducer model that is able to bias audio encoder features towards a provided list of words using a contextual biasing adapter [10]. CB-T therefore comprises four components: an audio encoder, label predictor, joint network, and CB adapter.

*Work done during an internship at Amazon Alexa.

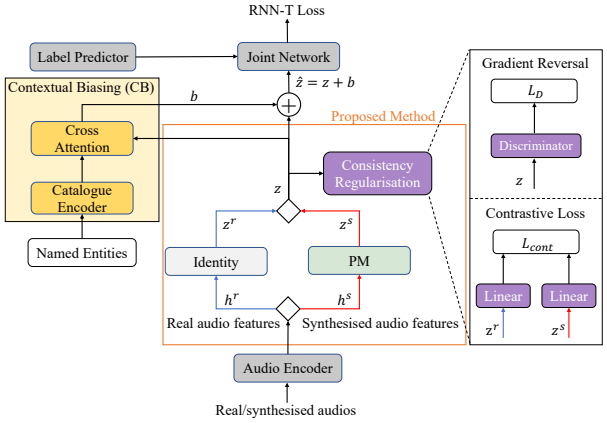


Figure 1: *An overview of CB-T models with PM for an input batch of one real and one synthesised utterance. The proposed PM, CB adapter, and basic components of RNN-T models are colour-coded green, yellow, and dark grey, respectively. Dark grey components are frozen during CB-T training.*

The CB adapter comprises two components: a catalogue encoder and a biasing adaptor. Given a catalogue of words or phrases, $\mathbf{C} = (c_1, c_2, ..., c_K)$, the catalogue encoder produces fixed-dimensional vector representations for each of the variable-length entities in the catalogue:

$$\mathbf{c}_i^e = \text{BiLSTM}(\text{Embedding}(\text{Tokenise}(c_i))), \quad (1)$$

where $\mathbf{c}_i^e$ is the last output of a BiLSTM encoder, and $\mathbf{C}^e = (\mathbf{c}_1^e, \mathbf{c}_2^e, ..., \mathbf{c}_K^e)$ is a list of encoded entities. Given RNN-T encoder features $\mathbf{h}_t$ at time t, the biasing adapter performs cross-attention [24] between $\mathbf{h}_t$ and $\mathbf{C}^e$ at each time step as follows:

$$\mathbf{b}_t = \text{Softmax}\left(\frac{\mathbf{h}_t\mathbf{W}^q(\mathbf{C}^e\mathbf{W}^k)^\intercal}{\sqrt{d_b}}\right)\mathbf{C}^e\mathbf{W}^v, \quad (2)$$

where $\mathbf{W}^q$, $\mathbf{W}^k$ and $\mathbf{W}^v$ are projection layers. Finally, contextualised features are the result of element-wise addition between encoded features and biasing vectors, $\hat{\mathbf{h}}_t = \mathbf{h}_t + \mathbf{b}_t$. The joint network takes contextualised features, $\hat{\mathbf{h}}_t$, and label predictor features as inputs and produces the posterior distribution over word-pieces. For readability we drop the time index t from future equations.

3. Methods

In this section we describe the proposed projection module and the two evaluated methods of consistency regularisation.

3.1. Projection module

The purpose of the Projection Module (PM) is to transform encoder features generated from synthesised audio such that they are as close as possible to those generated from real speech.

Given real audio $\mathcal{R}$ and synthesised audio $\mathcal{S}$, we obtain the corresponding encoder features $\mathbf{h}_\mathcal{R}$ and $\mathbf{h}_\mathcal{S}$ by feeding $\mathcal{R}$ and $\mathcal{S}$ through the RNN-T encoder. Transformed encoded features, $\mathbf{z}$, are then obtained as follows:

$$\mathbf{z} = \begin{cases} \mathcal{F}_{\text{PM}}(\mathbf{h}_\mathcal{S}) \\ \mathbf{h}_\mathcal{R}, \end{cases} \quad (3)$$

where $\mathcal{F}_{\text{PM}}$ denotes the PM function. $\mathbf{z}$ is then used as input to the CB adapter and added to the biasing vectors before being fed to the joint network (Fig. 1). The PM is trained jointly with the CB adapter. The rest of the RNN-T network is initialised using a pretrained model and frozen during training.

3.2. Consistency regularisation

Consistency regularisation penalises distinctiveness between encoder features of real and synthesised audio. Consistency regularisation and RNN-T losses are optimised jointly in a multi-task learning manner. In this work, dissimilarities are measured on utterance-level by applying time-axis averaging on real and synthesised audio encoder features. We explored two forms of consistency regularisation: gradient reversal and contrastive loss.

3.2.1. Gradient reversal

A domain discriminator is first added to the model with the task of classifying encoder states as coming from either real or synthetic audio. Gradient reversal (GR) then influences the PM to make features of real and synthesised audio indistinguishable by updating model weights in the direction that hinders the domain discriminator [22]. The discriminator is trained using the binary cross entropy loss, $\mathcal{L}_D$. Discriminator weights are updated as:

$$\theta_{\text{DISC}} = \theta_{\text{DISC}} - \alpha\partial\mathcal{L}_\mathcal{D}/\partial\theta_{\text{DISC}}, \quad (4)$$

where α is the learning rate and θ_{DISC} are the weights of the discriminator.

The PM is updated using gradients from RNN-T loss, $\mathcal{L}_{\text{RNN-T}}$, and scaled additive inverse of upstream gradients from the discriminator, i.e.:

$$\theta_{\text{PM}} = \theta_{\text{PM}} - \alpha(\partial\mathcal{L}_{\text{RNNT}}/\partial\theta_{\text{PM}} - \lambda\partial\mathcal{L}_\mathcal{D}/\partial\theta_{\text{PM}}), \quad (5)$$

where θ_{PM} are weights of the PM and λ is the scaling factor used to incorporate the reversed gradients.

3.2.2. Contrastive loss

Contrastive loss (CL) penalises dissimilarities between real and synthesised audio features that share the same transcripts [23].

Given real audio examples $\mathcal{R}$ in a training batch, we use the transcript of each of the real examples in $\mathcal{R}$ to synthesise paired TTS audio $\mathcal{S}$. We apply the contrastive loss to minimise the dissimilarities between $\mathcal{R}$ and $\mathcal{S}$ as shown in (6). The positive samples are pairs of transformed encoded features $\{(\mathbf{z}_\mathcal{R}^1, \mathbf{z}_\mathcal{S}^1), ..., (\mathbf{z}_\mathcal{R}^N, \mathbf{z}_\mathcal{S}^N)\}$ for real data $\mathcal{R}$ and synthesised audios $\mathcal{S}$, of which transcripts are the same, where N is the number of real examples. The negative samples are drawn from the other real audio examples in the batch and their paired TTS audio as shown in (7) and (8).

$$\mathcal{L}_{cont} = -\sum_{i=1}^{N}\log\frac{\exp(sim(\mathcal{G}_r(\mathbf{z}_\mathcal{R}^i), \mathcal{G}_p(\mathbf{z}_\mathcal{S}^i))/\tau)}{\text{denom}_r + \text{denom}_s} \quad (6)$$

$$\text{denom}_r = \sum_{k=1:k\neq i}^{N}\exp(sim(\mathcal{G}_r(\mathbf{z}_\mathcal{R}^i), \mathcal{G}_r(\mathbf{z}_\mathcal{R}^k))/\tau) \quad (7)$$

$$\text{denom}_s = \sum_{k=1k\neq i}^{N}\exp(sim(\mathcal{G}_r(\mathbf{z}_\mathcal{R}^i), \mathcal{G}_p(\mathbf{z}_\mathcal{S}^k))/\tau) \quad (8)$$

where $\mathbf{z}_\mathcal{R}^i$ and $\mathbf{z}_\mathcal{S}^i$ are transformed encoded features of the i^{th} real audio and its synthesised paired audio, averaged over time-axis. The function $\mathcal{G}_r$ and $\mathcal{G}_p$ are projection layers dedicated

Table 1: *Statistics for real-speech corpora as more utterances are replaced by synthetic speech.* (X, Y) *represent statistics for real and synthesised subsets within each data splits, respectively.*

m (%)	data size (hr)	#utterances (k)	#unique names (k)
100	(185, 0)	(219, 0)	(52, 0)
70	(117, 36)	(138, 78)	(36, 15)
50	(86, 53)	(101, 113)	(26, 26)
30	(53, 70)	(63, 150)	(15, 36)
10	(18, 89)	(21, 190)	(5, 46)
0	(0, 99)	(0, 210)	(0, 52)

for real and synthesised paired audio, respectively. We follow [23, 25] and use cosine similarity as the distance function sim.

4. Experimental Setup

4.1. Training datasets

We used an in-house American English voice assistant dataset with each utterance consisting of audio, transcript and a catalogue of named entities. The training data is not associated with identifying information. The real traffic data contains 219k utterances (185 hours) and 52k unique named entities. We randomly selected $m\%$ of the unique named entities and reserved the transcripts containing the selected named entities for real audios. The real audios paired with the transcripts containing the remaining $100 - m\%$ unique named entities were replaced by synthesised audios. As a result, we had real and synthesised corpora that did not have named entities in common. The details for corpora with different values of m are depicted in Table 1.

In some experiments we used an extended corpus which includes all the available real audio ($m = 100\%$) together with 20k synthetic utterances generated from new synthetic text: such text was generated by randomly replacing named entities in each of the existing real transcript with a new entity sampled from a list of approximately 10M named entities from the same domain. The contextual catalogues of such synthetic utterances were created by randomly sampling N_c examples from the same list of named entities, where the catalogue size, N_c, was sampled from the distribution of catalogue lengths observed in real traffic. We obtained 144 hours of synthesised audios after performing one pass of the extension procedure. We repeated the procedure four times to create extended corpora containing 144, 288, 432, and 578 hours. The four extended corpora had 235k, 460k, 675k, and 885k unique names, respectively.

To synthesise audio we fed pairs of transcriptions and randomly selected speaker profiles, drawn from a pool of one thousand speakers, to an in-house TTS model [26].[1]

4.2. Evaluation

We evaluated our models on real-speech test sets consisting of general data (16k utterances) and data containing named entities (NE, 19k utterances). Where utterances contained named entities, we ensured that their contextual catalogue included that named entity.

Relative word error rate reduction (WERR) is reported for the general test set. WERR is defined as $\text{WERR} = (\text{WER}_B - \text{WER}_A)/\text{WER}_B$, where B is the baseline, and A is the model under evaluation. For NE, we report relative name-entity word error rate reduction (NE-WERR). For both metrics, positive values indicate improvements. Unless explicitly stated, the baseline used to compute the relative error rate reduction is the CB-T trained with all real audio, representing the expected upper bound.

[1] Synthesised audios were generally shorter than real audios and not every transcript was successfully synthesised by the TTS model, resulting in fewer hours of synthesised speech.

4.3. Architecture details

The base RNN-T model had 150M parameters in total. The audio encoder consisted of eight LSTM layers, with a time-reduction factor of two applied after the first two layers. The label predictor had two LSTM layers. We used 1280 hidden units in each LSTM layer of the audio encoder and label predictor. The joint network was an element-wise addition, after which a projection to the output layer of size 4001 was applied (4000 word-pieces plus blank state). The catalogue encoder in the CB adapter consisted of a bidirectional LSTM layer with 64 hidden units in each direction. The attention layer of the CB adapter used one attention head and projected queries and keys to 128 dimensions, with values projected to 1280 dimensions to match the output of the audio encoder. The catalogue size was set to a maximum of 300 during training to fit within memory. For evaluation, we increased the maximum catalogue size to 5000. We performed beam search with a beam size of eight. Decoding results are from single-pass decoding; we did not make use of an external language model.

In terms of PM architecture, we experimented with feed-forward neural networks and transformer encoders. The feed-forward PM (FPM) had two fully-connected layers with a ReLU activation function applied to the output of the first layer. Transformer PM (TPM) was a single Transformer encoder with four attention heads [27]. Inputs and outputs of the PM were 1280 dimensional vectors in both cases. The domain discriminator used for gradient reversal had one 128 dimensional feed-forward layer with ReLU activation. Dropout at a rate of 0.5 was applied to this layer. The projection functions used for the contrastive loss, $\mathcal{G}_r$ and $\mathcal{G}_p$, comprised two 1280-dimensional linear layers with ReLU activation.

4.4. Training details

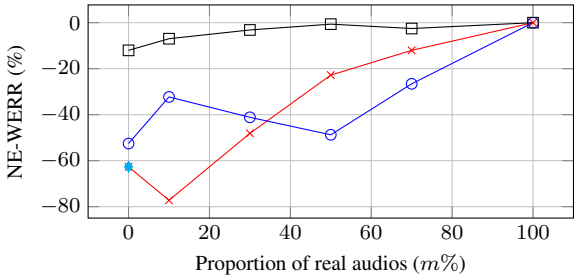
The weights of the audio encoder, label predictor, and joint network were pretrained using 160k hours of real speech. The CB adapter and PM were jointly trained from scratch in a second training stage for 100k steps, while the rest of the network was kept frozen. We used Adam optimiser [28] with a static learning rate of 8e-4. We used 16 GPUs to train the models. Unless stated otherwise, we used m and $100 - m$ as weights for real and synthesised datasets, i.e. each batch comprised $m\%$ of real audio and $100 - m\%$ of synthetic audio.

We used $\lambda = 0.1$ for gradient reversal and $\tau = 0.07$ for contrastive loss to weight the consistency regularisation during training.

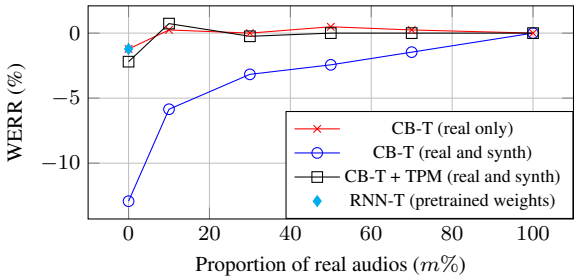
5. Experimental Results

5.1. Replacing real speech with synthetic speech

We begin by comparing CB-T with CB-T+TPM for different proportions of real audio m in Figure 2. We trained CB-T and CB-T+TPM models using two training sets for each value of m: the $m\%$ real audios (real only), and both real and synthesised audios (real and synt). As shown in Figure 2a, introducing syn-



(a) *NE-WERR (%) on named entity test set*



(b) *WERR (%) on general test set*

Figure 2: *Performance on the real-speech corpora with varying amount of names reserved for real audios. For each split, we replaced* $100 - m\%$ *of training utterances with synthesised audios. The "real only" models were trained using the* $m\%$ *of real audios only. "pretrain" denotes the performance of the baseline RNN-T model without CB.*

thesised audios to the training data without using PM degraded the NE-WERR when $m > 30\%$. The proposed CB-T+TPM is found to consistently outperform CB-T across every split in terms of NE-WERR, illustrating how the projection module has effectively reduced the mismatch introduced by the synthetic audio, allowing us to leverage such audio when training contextual adapters. Performance of the baseline is maintained with $m = 50\%$, and degrades by only 12% if all real audio is replaced by synthetic audio ($m = 0$). Introducing synthetic audio without compensating for the mismatch was found to degrade performance on general data (Figure 2b), likely due to poor quality embeddings caused by training the catalogue encoder with uncompensated synthetic audio resulting in spurious matches in the attention layer. In contrast, error rates on general data are largely stable after introducing TPM.

Table 2 presents the comparison between models for the data split of $m = 10\%$. The CB-T model (M2) is significantly worse as it only had access to 10% of the real audio during training. We introduced synthesised audios to the CB-T model in order to compensate the missing 90% of real utterances (M3). Even though synthesised utterances improved the NE-WERR on the named entity test set, we observed degradation on general data. Adding PM to CB-T substantially improved the error rates for both test sets. Although WERR on General of both FPM and TPM were on par with M1, the more complex TPM outperformed the FPM on named entities. Consistency regularisation further enhanced TPM. The models trained using gradient reversal had only 3% degradation to M1 despite having access to only 10% of the real data. The contrastive loss also helped, but was not quite as effective as GR.

Table 2: *Error rate reductions for the corpus where 90% of real audio is replaced by synthetic speech. GR and CL stand for gradient reversal and contrastive loss, respectively.*

Model	data size (hr) (real, synt)	Named Entities (%NE-WERR)	General (%WERR)
M1) CB-T	(185, 0)	0.0	0.0
M2) CB-T	(18, 0)	-79.6	0.2
M3) CB-T	(18, 89)	-34.3	-7.3
M4) CB-T + FPM	(18, 89)	-12.1	0.5
M5) CB-T + TPM	(18, 89)	-6.6	**0.7**
M6) CB-T + TPM + GR	(18, 89)	**-3.1**	**0.7**
M7) CB-T + TPM + CL	(18, 89)	-4.6	0.2

Table 3: *Performance evaluation for different sizes of extended corpora.*

Model	data size (hr) (real, synt)	Named Entities (%NE-WERR)	General (%WERR)
CB-T	(185, 0)	0.0	0.0
CB-T	(185, 144)	-58.5	-2.0
CB-T + TPM	(185, 144)	12.3	0.5
CB-T + TPM + GR	(185, 144)	**16.7**	-0.5
CB-T + TPM + CL	(185, 144)	15.5	0.0
CB-T + TPM + GR	(185, 288)	16.1	-0.2
CB-T + TPM + GR	(185, 432)	15.0	-0.2
CB-T + TPM + CL	(185, 578)	16.1	0.0

5.2. Extending the real corpus with synthetic audio

Finally, we investigated whether the proposed method could be used to improve performance over the baseline CB-T trained with the full real audio corpus by introducing extending corpora of synthetic audio during training (Section 4.1). Equal weighting was applied to the real and extended corpora.

As shown in Table 3, the best system is CB-T+TPM+GR. We found that NE-WERR can be further improved, by over 16%, by adding 144 hours of synthetic audio. The approach has limitations, however; increasing the size of the extended corpus from 144 to 578 hours did not yield any further improvements. We hypothesise two potential reasons behind this limitation: i.) sentence and speaker diversity is relatively limited, with no further variety in carrier phrases being introduced in the synthesised transcripts and only 1000 speaker profiles being used to generate the audio, and ii.) we may have reached the upper bound of performance for the model architecture, where even further real audio would not improve performance, however we were unable to evaluate this hypothesis due to no further real data being available.

6. Conclusions

In this work we have shown how a projection module can be used to reduce the mismatch between real and synthetic audio, allowing contextual biasing adapters to be trained with synthetic audio. We demonstrated how the proposed approach can be used to achieve on-par performance with a model trained using the full real audio training corpora when 50% of the real audio was replaced with synthesised speech. We also showed how performance of the model can be further improved by extending the training set with additional synthesised examples. Furthermore, our proposed method has no impact on the computational cost of inference and is suitable for streaming use-cases.

7. References

[1] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented Transformer for Speech Recognition," in *Proceedings of Interspeech*, 2020.

[2] A. Graves, "Sequence Transduction with Recurrent Neural Networks," in *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2012.

[3] A. Zeyer, K. Irie, R. Schlüter, and H. Ney, "Improved training of end-to-end attention models for speech recognition," in *Proceedings of Interspeech*, 2018.

[4] D. Zhao, T. N. Sainath, D. Rybach, P. Rondon, D. Bhatia, B. Li, and R. Pang, "Shallow-Fusion End-to-End Contextual Biasing." in *Proceedings of Interspeech*, 2019.

[5] A. Gourav, L. Liu, A. Gandhe, Y. Gu, G. Lan, X. Huang, S. Kalmane, G. Tiwari, D. Filimonov, A. Rastrow *et al.*, "Personalization strategies for end-to-end speech recognition systems," in *Proceedings of ICASSP*, 2021, pp. 7348–7352.

[6] R. Cabrera, X. Liu, M. Ghodsi, Z. Matteson, E. Weinstein, and A. Kannan, "Language model fusion for streaming end to end speech recognition," *arXiv preprint arXiv:2104.04487*, 2021.

[7] M. Jain, G. Keren, J. Mahadeokar, G. Zweig, F. Metze, and Y. Saraf, "Contextual RNN-T For Open Domain ASR," in *Proceedings of Interspeech*, 2020.

[8] F.-J. Chang, J. Liu, M. Radfar, A. Mouchtaris, M. Omologo, A. Rastrow, and S. Kunzmann, "Context-aware Transformer Transducer for Speech Recognition," in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, 2021.

[9] G. Pundak, T. N. Sainath, R. Prabhavalkar, A. Kannan, and D. Zhao, "Deep Context: End-to-End Contextual Speech Recognition," in *Proceedings of the IEEE Workshop on Spoken Language Technology*, 2018.

[10] K. M. Sathyendra, T. Muniyappa, F.-J. Chang, J. Liu, J. Su, G. P. Strimel, A. Mouchtaris, and S. Kunzmann, "Contextual Adapters for Personalized Speech Recognition in Neural Transducers," in *Proceedings of ICASSP*, 2022.

[11] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang, Q. Liang, D. Bhatia, Y. Shangguan, B. Li, G. Pundak, K. C. Sim, T. Bagby, S.-Y. Chang, K. Rao, and A. Gruenstein, "Streaming End-to-end Speech Recognition For Mobile Devices," in *Proceedings of ICASSP*, 2019.

[12] C. Peyser, H. Zhang, T. N. Sainath, and Z. Wu, "Improving Performance of End-to-End ASR on Numeric Sequences," in *Proceedings of Interspeech*, 2019.

[13] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis," in *NeurIPS*, 2020.

[14] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. J. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions," in *Proceedings of ICASSP*, 2018.

[15] A. van den Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. van den Driessche, E. Lockhart, L. Cobo, F. Stimberg, N. Casagrande, D. Grewe, S. Noury, S. Dieleman, E. Elsen, N. Kalchbrenner, H. Zen, A. Graves, H. King, T. Walters, D. Belov, and D. Hassabis, "Parallel WaveNet: Fast High-Fidelity Speech Synthesis," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

[16] T.-Y. Hu, M. Armandpour, A. Shrivastava, J.-H. R. Chang, H. Koppula, and O. Tuzel, "SYNT++: Utilizing Imperfect Synthetic Data to Improve Speech Recognition," in *Proceedings of ICASSP*, 2022.

[17] X. Zheng, Y. Liu, D. Gunceler, and D. Willett, "Using Synthetic Audio to Improve The Recognition of Out-Of-Vocabulary Words in End-To-End ASR Systems," in *Proceedings of ICASSP*, 2021.

[18] Z. Chen, Y. Zhang, A. Rosenberg, B. Ramabhadran, P. Moreno, and G. Wang, "Tts4pretrain 2.0: Advancing the use of Text and Speech in ASR Pretraining with Consistency and Contrastive Losses," in *Proceedings of ICASSP*, 2022.

[19] G. Wang, A. Rosenberg, Z. Chen, Y. Zhang, B. Ramabhadran, Y. Wu, and P. Moreno, "Improving Speech Recognition Using Consistent Predictions on Synthesized Speech," in *Proceedings of ICASSP*, 2020.

[20] A. Fazel, W. Yang, Y. Liu, R. Barra-Chicote, Y. Meng, R. Maas, and J. Droppo, "SynthASR: Unlocking Synthetic Data for Speech Recognition," in *Proceedings of Interspeech*, 2021.

[21] G. Kurata, G. Saon, B. Kingsbury, D. Haws, and Z. Tüske, "Improving Customization of Neural Transducers by Mitigating Acoustic Mismatch of Synthesized Audio," in *Proceedings of Interspeech*, 2021.

[22] Y. Ganin and V. Lempitsky, "Unsupervised Domain Adaptation by Backpropagation," in *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015.

[23] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A Simple Framework for Contrastive Learning of Visual Representations," in *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

[24] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," in *ICLR*, 2015.

[25] D. Jiang, W. Li, M. Cao, W. Zou, and X. Li, "Speech SIMCLR: Combining Contrastive and Reconstruction Objective for Self-supervised Speech Representation Learning," in *Proceedings of Interspeech*, 2021.

[26] I. Vallés-Pérez, J. Roth, G. Beringer, R. Barra-Chicote, and J. Droppo, "Improving multi-speaker TTS prosody variance with a residual encoder and normalizing flows," in *Proceedings of Interspeech*, 2021.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," in *NuerIPS*, 2017.

[28] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *ICLR*, 2015.