

Residual Adapters for Parameter-Efficient ASR Adaptation to Atypical and Accented Speech

Katrin Tomanek*, Vicky Zayats*, Dirk Padfield, Kara Vaillancourt, Fadi Biadisy

Google, USA

{katrintomanek, vzayats, padfield, kvailla, biadisy}@google.com

Abstract

Automatic Speech Recognition (ASR) systems are often optimized to work best for speakers with canonical speech patterns. Unfortunately, these systems perform poorly when tested on atypical speech and heavily accented speech. It has previously been shown that personalization through model fine-tuning substantially improves performance. However, maintaining such large models per speaker is costly and difficult to scale. We show that by adding a relatively small number of extra parameters to the encoder layers via so-called residual adapter, we can achieve similar adaptation gains compared to model fine-tuning, while only updating a tiny fraction (less than 0.5%) of the model parameters. We demonstrate this on two speech adaptation tasks (atypical and accented speech) and for two state-of-the-art ASR architectures.

1 Introduction

Automatic Speech Recognition (ASR) systems have achieved great success on a diverse set of acoustic and linguistic conditions, domains and speech patterns. State-of-the-art ASR systems are typically trained on tens of thousands of hours of speech data, and they perform well as long as these domains and conditions are well represented in the training data.

Understandably, the distribution of such data typically focuses on the canonical *and* typical spoken language patterns of the target language, i.e., regional dialects, common accents and frequent non-native accents. As a result, these systems may perform poorly on the tail of the distribution which may include “heavily” accented speech and/or speech with atypical speech patterns (Darley et al., 1975). Atypical speech includes dysarthric speech, speech impairments (due to, for example, ALS, stroke, traumatic brain injury, down syndrome, cerebral palsy, and MS), stuttering, deaf

speech, or severe hyper-nasality due to cleft lip and palate. The lack of sufficient training data for these accents and atypical speech in the training distribution may result in a poor experience for a large segment of the population, leaving the less fortunate communities behind when it comes to speech-enabled technologies (Moore et al., 2018).

Studies on accented speech showed word error rates (WER) twice or three times as high for accented speech compared to the more standard US accent (Sainath et al., 2020; Ghorbani and Hansen, 2018). Even worse performance is observed for speakers with speech impairments (Moore et al., 2018). Our goal, in this paper, is to efficiently build scalable models that can adapt to non-canonical or atypical speech.

It has been shown that speech models originally developed for typical speech can be successfully fine-tuned with limited amounts of data to accented or impaired speech (Zhu et al., 2019; Shor et al., 2019; Gale et al., 2019; Mustafa et al., 2014; Biadisy et al., 2019; Doshi et al., 2020; Green et al., 2021). Nevertheless, one major challenge with adapting models to either individuals or small groups of speakers is that it is necessary to scale the number of models that need to be maintained and hosted. For example, for smart devices powerful enough to run ASR models on-device, having to deploy and store an additional (potentially large) model may take up valuable on-device resources. Similarly, providing personalized models for a large population of speakers in a centralized/server-based scenario is not feasible.

We propose to mitigate this issue by injecting residual adapter layers into the architecture. Particularly, we use a bottleneck architecture that requires a tiny number of parameters ($< 0.5\%$ in our scenario) compared to the full model update via fine-tuning. Then, while keeping the original pre-trained model parameters frozen, we update only the parameters of the adapter layers as we train

* Equal contribution

on the custom data of interest. This provides an easy way to deploy and store adapted models: a (generic) base model is deployed to all clients, and each individual or group can receive a personalized set of trained adapter layers that is small in size.

The main contributions of the paper are as follows. We show that residual adapters work extremely well for acoustic adaptation of different speech models. We present extensive experiments with adapter layers in two very different ASR use-cases: personalized models for atypical speech, and group models for accented speech. We also demonstrate that adapter layers work well in two different, state-of-the-art end-to-end ASR architectures, Neural Network Transducers (RNN-T), and Transformer Transducers (T-T). This emphasizes the flexibility of this approach and its suitability as a standard alternative to full fine-tuning for arbitrary models. Our results clearly demonstrate how adaptation via adapter layers solves the issue of parameter inefficiency while largely retaining the significant adaptation gains achievable through model adaptation with in-domain data.

2 Related Work

Model fine-tuning has been successfully applied to domain adaptation for a variety of NLP tasks (Devlin et al., 2019; Sun et al., 2019), Machine Translation (Freitag and Al-Onaizan, 2016), and speech recognition and conversation systems, including accented and atypical speech (Zhu et al., 2019; Shor et al., 2019; Gale et al., 2019; Biadys et al., 2019; Doshi et al., 2020; Green et al., 2021)

A major disadvantage of model fine-tuning is its parameter inefficiency since it re-trains all (or a large portion of) the model parameters on given task- or domain-specific data, resulting in a copy of the model for that task/domain. This is especially problematic for personalization of models due to the resulting high number of specialized models.

Concatenating input features and speaker-dependent vectors, such as i-vectors, is a parameter-efficient speaker adaptive approach that has been applied to both acoustic models as well as end-to-end ASR models (Saon et al., 2013, 2021). However, only moderate improvements have been achieved, even on typical speech. We speculate that such a *static*, low dimensional representation may not be sufficient to capture the complex acoustic-phonetic patterns (e.g., consonant dropping and vowel dropping, extreme vowel reduction or length-

ening, missing phonemes and even syllables, very irregular speaking rate and rhythm) often found in impaired speech.

Residual adapters were originally introduced by Rebuffi et al. (2017) for computer vision tasks as an alternative to fine-tuning. These first residual adapter modules consisted of a single projection layer added between layers of a pre-trained network. Houlisby et al. (2019) proposed a variation consisting of a bottleneck structure (down-projection through feed forward layer, RELU, up-projection) for task-specific adaptation of BERT models. Adapter modules were added after each sub-layer within a transformer layer, and the weights of the residual adapters as well as existing layer normalization parameters were updated during training. Finally, Bapna and Firat (2019) have formulated a simplification of residual adapters in the context of domain-adaptation for Machine Translation. Each residual adapter module has its own layer normalization block, followed by a down- and up-projection feed forward network. They argued that by including layer normalization in the residual adapter block, these modules are plug-able into arbitrary blocks of pre-trained modules because they learn the activation pattern of the layer into which they are injected.

Kannan et al. (2019) proposed to use adapters on top of multi-lingual ASR models to further improve their performance (they report up to 9% WER improvement for some of the 5 languages of the multi-lingual model). Our focus is different in that we consider residual adapters in speech personalization scenarios where the number of adapted models is several orders of magnitudes higher (e.g., tens of thousand of speakers with atypical speech and potentially hundreds of accents and dialects) and also not static (e.g. speech impairments often progress over time).

Learning Hidden Unit Contribution (LHCU) (Swietojanski et al., 2016) is another approach to more parameter efficient speaker adaptation. Instead of updating all weights of a model, LHUC adds learned factors to the output of each hidden unit modulating their amplitude. However, Bapna and Firat (2019) have shown that using residual adapters is much more effective.

3 Methods

For our experiments, we chose two state-of-the-art end-to-end ASR architectures: the Recurrent Neu-

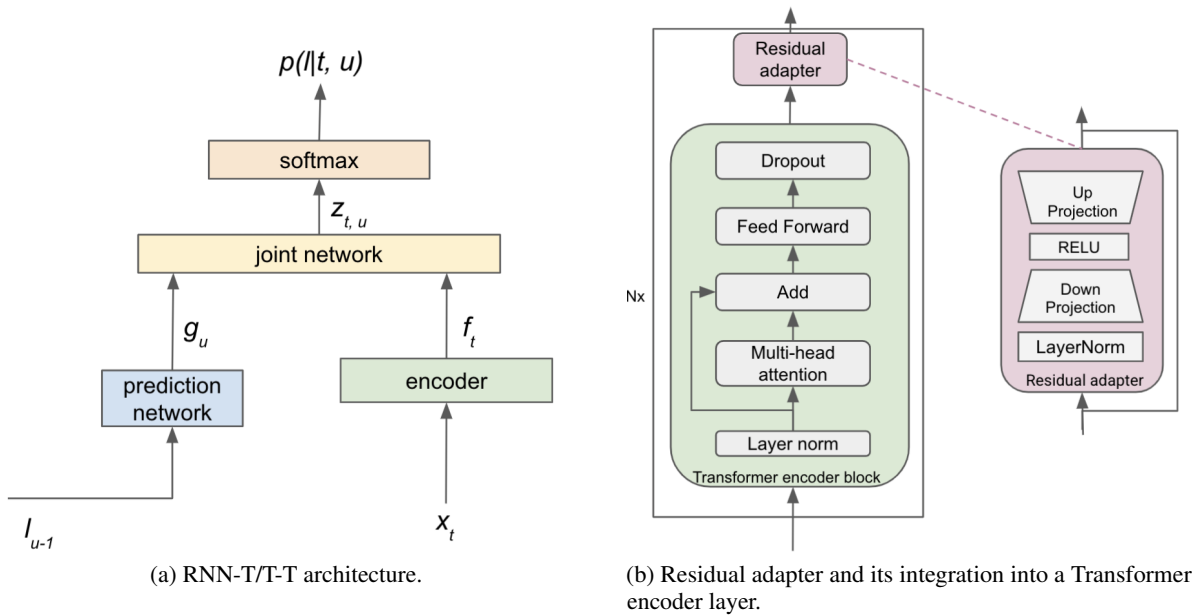


Figure 1: Overview of RNN-T and T-T architectures and residual adapter module.

ral Network Transducers (RNN-T) (Graves et al., 2013; He et al., 2019; Sainath et al., 2020) and the Transformer Transducer (T-T) (Zhang et al., 2020). Both architectures enable deployment on mobile devices, support streaming (He et al., 2019), and have demonstrated high performance.

Both architectures consist of three main components: an encoder, a prediction network that incorporates label history and serves as a language model component (decoder), and a joint layer that combines predictions made by the encoder and the prediction network and feeds into a softmax. All components of the two architectures are identical except the encoder stack. The prediction network consists of 2 uni-directional LSTM layers. Inputs are 128-dimensional log Mel features computed every 10 milliseconds. 4 consecutive features are stacked with a stride of 3 frames to yield a 512-dimensional input to the encoder every 30 milliseconds. Our output vocabulary consists of 4096 word piece tokens. Figure 1a shows a high-level overview of both architectures.

For RNN-T, the encoder consists of 8 LSTM layers; for T-T, we use 15 Transformer layers in the encoder. Both architectures are trained with the RNN-T loss (Bagby et al., 2018). To make T-T streamable, the attention calculation pays attention to past contexts only, which makes this architecture analogous to a uni-directional RNN.

We propose to utilize residual adapter modules as outlined by Bapna and Firat (2019) for our

adaptation approach. Each residual adapter block starts with layer normalization applied to the inputs, followed by a feed-forward layer with down-projection to dimension d_b , a non-linear activation (RELU), and another feed-forward layer with up-projection to the original input dimension d_i . All weights of the residual adapter module are randomly initialized.

Figure 1b shows such a residual adapter module and its integration within the Transformer encoder. We add residual adapters to each encoder layer, resulting in 8 adapter layers for RNN-T and 15 adapter layers for T-T. The bottleneck dimension d_b enables control of the number of parameters of each residual adapter module and thus the capacity available during adaptation.

4 Experiments

We analyze the performance of residual adapters as an alternative to model fine-tuning in two scenarios: adaptation to (a) atypical speech and (b) accented speech. For atypical speech, we build a personalized, speaker-dependent model for each speaker based on their data. For accented speech, we build per-accent models (i.e. speaker-independent models) and also experiment with a multi-accent adaptation scenario where one model is used for all covered accents. We conduct experiments using both ASR transducer architectures (RNN-T and T-T).

4.1 Accented Speech Dataset

For the accented speech adaptation task, we use Mozilla’s Common Voice corpus (v5.1) (Ardila et al., 2020). It contains spoken utterances of users reading sentences. Recordings were verified by other contributors using a simple voting system. While the full corpus contains 60 languages, for this work we use a subset containing only English recordings. We make use of Common Voice’s metadata to extract accent information and use all 10 accents with more than 1k recordings, including (in order of decreasing number of recordings): England (en), India (in), Australia (au), Canada (ca), Scotland (sc), Ireland (ir), New Zealand (nz), Africa (af), Singapore (si), and Philippines (ph).

We randomly split all utterances from each accent into train/dev/test subsets. The resulting subset sizes per accent are shown in Table 5. Table 1 shows utterance counts and length (in words and seconds) aggregated across all accents.

4.2 Atypical Speech Dataset

We use the Euphonia corpus (MacDonald et al., 2021) for the atypical speech personalization task. This corpus consists of over 1 million utterance recordings of over 1000 anonymized speakers with different types and severity levels of speech impairments. Similar to the Common Voice corpus, all recordings in the Euphonia corpus are prompted speech. All our experiments are performed on a random subset of 100 speakers who have each recorded more than 1000 utterances. The resulting subset is very diverse, covering speakers with 15 different etiologies (31% with amyotrophic lateral sclerosis (ALS), 20% Down Syndrome, 14% cerebral palsy, 6% Parkinson’s Disease, 5% hearing impairment etc) and different speech impairment severity levels (47% mild, 32% moderate, 21% severe). We use the predefined per-speaker train, dev, and test splits (80%/10%/10%).

Table 1 shows utterance counts and length (in words and seconds) aggregated across all 100 speakers. Note that speakers with a speech impairment often have a lower speaking rate and frequently pause between individual words and before speaking. This is reflected in the relatively low ratio of words per second in the Euphonia corpus.

4.3 Experimental Settings

We follow a similar fine-tuning recipe as described in Green et al. (2021). We start from a speaker-

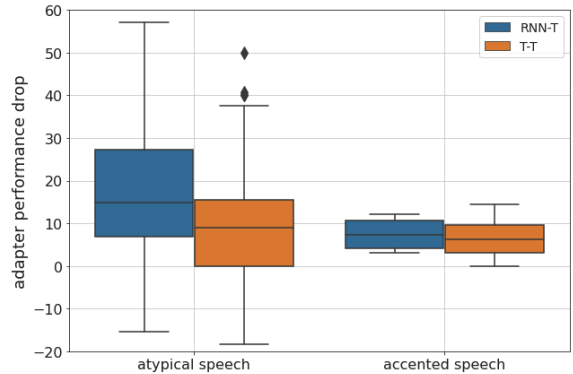


Figure 2: Distribution of adapter performance drop across all speakers/accents.

independent base model pre-trained on 162k hours of typical (mostly American English) speech. This base model has been optimized to (a) be robust across various application domains and acoustic conditions, and (b) generalize well to unseen conditions (Narayanan et al., 2019). The same base model is used across all of our experiments.

We use SpecAugment (Park et al., 2019) for data augmentation, limit training to a maximum of 50k steps (atypical speech) and 30k steps (accented speech) and employ small batch sizes (32 for atypical speech, 256 for accented speech with RNN-T, and 128 for accented speech with T-T).

We only update the weights of the encoder layers, as our focus is on learning acoustic-phonetic variability as opposed to vocabulary and language variability. Accordingly, weights of the joint layer and the prediction network are always kept frozen. When training with residual adapters, we freeze *all* parameters of the base model and only update the residual adapter layers. Table 2 shows the resulting number of parameters updated for the different adaptation strategies. For example, residual adapters with a bottleneck dimension of 16 yield more than $100\times$ parameter reduction compared to the encoder fine-tuning scenario.

Word error rate (WER) is measured on the respective test splits. The best checkpoints are chosen based on the WER on the dev split.

5 Results

Table 2 compares fine-tuning versus residual adapters across both tasks and architectures.¹ Unless otherwise specified, we report per-accent (as

¹The performance of RNN-T models is slightly better than T-T across all of our experiments despite the bigger encoder size of T-T. However, T-T’s training time is much shorter.

Dataset	Subset	Hours	# utts	Words per utt mean (std)	Seconds per utt mean (std)
Euphonia	home automation	73	69.3k	3.2 (1.4)	3.8 (2.0)
	conversational	67	37.1k	7.4 (4.0)	6.5 (4.0)
Common Voice	10 accents	183	118.6k	10.3 (2.8)	5.6 (1.6)

Table 1: Utterance length statistics in number of words and seconds per dataset.

Arch	Adaptation style	Updated params		Atypical Speech		Accented Speech	
		Total	Relative	WER	Relative	WER	Relative
RNN-T	Unadapted	–	–	35.6	–	19.9	–
	Fine-tune, full enc	98.7M	81%	6.0	80%	13.9	29%
	Fine-tune, enc layer 1	10.8M	9%	10.9	61%	15.8	18%
	Fine-tune, enc layers 1-3	39.6M	32%	6.9	75%	13.4	28%
	Residual Adapters, $b_d = 16$	197K	<0.2%	6.8	77%	14.1	24%
T-T	Unadapted	–	–	38.4	–	21.6	–
	Fine-tune, full enc	144.3M	85%	6.1	78%	13.2	35%
	Fine-tune, enc layer 1	9.6M	6%	10.8	60%	16.3	22%
	Fine-tune, enc layers 1-3	28.9M	17%	8.4	72%	14.8	29%
	Residual Adapters, $b_d = 16$	507K	<0.5%	7.1	75%	14.1	31%

Table 2: Aggregated overview of adaptation results for both tasks. The number of updated parameters is given as well as the percentage of the total; the total number of parameters is about 122M for RNN-T and about 168M for T-T. For atypical speech, we report median WER across all 100 speakers. For accented speech, we report mean WER across 10 accents for the per-accent adaptation scenario. The percentages in the WER columns are the relative WER improvement γ (Eq. 1) over the unadapted model.

opposed to multi-accent) adaptation results. Adaptation performance is compared with performance on the *unadapted* base model. In addition to WERs, we also report the *relative WER improvement* over the unadapted model:

$$\gamma = \frac{\text{WER}(\text{unadapted}) - \text{WER}(\text{adapted})}{\text{WER}(\text{unadapted})} \quad (1)$$

For residual adapters, we identified the best learning rate and bottleneck dimensions during hyper-parameter tuning on the dev set (see Section 5.1). In addition to comparing residual adapters to a scenario where we fine-tune the entire encoder, we also test the impact of fine-tuning only a few layers (1-3) of the encoder. However, this alternative for reducing the number of updated parameters is less efficient than residual adapters, which have a much lower parameter footprint due to their bottleneck architecture.

Table 2 shows that on the atypical speech personalization task adapting the *full* encoder per speaker, we observe a relative reduction of 80% in median WER across speakers for RNN-T.² However, this strategy requires 81% of the model parameters to

²Green et al. (2021) report similar improvements over 500

be updated and stored per speaker. Using residual adapters, on the other hand, we achieve relative WER reduction of 77% across all speakers for RNN-T. Although fine-tuning is slightly better than adaptation with residual adapter layers, the latter only needs to update about 0.2% of the parameters. We observe similar trends with T-T.

Comparing to a scenario where we update only a few bottom layers of the encoder, we observe a significant³ WER increase compared to full encoder fine-tuning. Residual adapters, while using less than 0.2% of the parameters, perform significantly better than updating only the first encoder layer (9% of parameters on RNN-T) and slightly better than updating the encoder layers 1-3 (32% of parameters on RNN-T).

For accented speech, Table 2 shows that fine-tuning leads to more moderate improvements of 29% for RNN-T and 35% for T-T (averaged across all accents). Similar to the personalization task,

speakers of the Euphonia corpus; their in-depth analysis shows that this holds across different severities and types of speech impairment.

³Throughout this paper, we use paired t-tests to measure statistical significance (indicated as significant for p-values < 0.05)

residual adapters performs slightly worse than fine-tuning (24% improvement for RNN-T and 31% for T-T), but require the update of only a fraction of parameters. The alternative of updating only the first encoder layer shows a much poorer performance.

Figure 2 shows the *adapter performance drop*, the relative WER reduction when switching from fine-tuning to residual adapters calculated per speaker/accent (lower is better):

$$\delta = \frac{\text{WER}(\text{adapters}) - \text{WER}(\text{fine-tuned})}{\text{WER}(\text{fine-tuned})} \quad (2)$$

T-T exhibits a lower average adapter performance drop δ compared to RNN-T on both tasks. This is likely due to the higher overall capacity of residual adapters when applied to T-T due to the higher number of encoder layers (15 encoder layers for T-T, 8 for RNN-T).

5.1 Hyper-Parameter Tuning

The results reported in Table 2 are for bottleneck dimension and learning rates found to work well in hyper-parameter tuning experiments where we ran a grid search over a combination of the two. For the learning rate, we evaluated ($1e - 5$, $1e - 4$, $1e - 3$, $1e - 2$), and for the bottleneck dimension, we evaluated 4, 16, 32, 128. We use a random subset of 20 speakers for the atypical speech task for parameter tuning to make search feasible; for accents, search was run across all 10 accents.

For the atypical speech personalization task, a learning rate of $1e - 5$ worked best for fine-tuning, and $1e - 3$ for residual adapters (both on RNN-T and T-T).⁴ For accents, we found the best learning rate for fine-tuned models using RNN-T to be $1e - 5$, and $1e - 4$ for T-T. For residual adapters, the best learning rate was $1e - 4$ (RNN-T) and $1e - 3$ (T-T). We found that accents with high amounts of training data tended to be more tolerant to higher learning rates compared to accents with limited training data.

Overall, these experiments show that adapters require on average a learning rate one order of magnitude higher than fine-tuning the encoder for both T-T and RNN-T. This may be attributable to the much smaller capacity of adapters and/or to their random initialization.

⁴For individual speakers, higher or lower learning rates might individually lead to better performance. However, in a practical personalization scenario with hundreds of speakers, such tuning is impossible, so we chose a one-size-fits-all approach.

For the bottleneck dimension, we found that $b_d = 4$ often leads to a higher adapter performance drop on the atypical speech task, although for some speakers – especially those with mild impairment and generally relatively low (≤ 25) WER on the unadapted models – even this bottleneck dimension worked very well. A bottleneck dimension of $b_d = 128$ rarely led to increased performance over $b_d = 16$ and $b_d = 32$. Between the latter two, we could not make out a clear pattern; they often performed equally well. For accented speech, when adapting for individual accents, we found that both $b_d = 4$ and $b_d = 16$ achieve similar performance. A bottleneck dimension $b_d = 128$ on average leads to worse performance than $b_d = 16$. Given these results, we chose a bottleneck dimension of $b_d = 16$ for all reported experiments, unless otherwise noted.

5.2 Atypical Speech Personalization

In this section we further analyze performance of the residual adapters for the atypical speech personalization task, zooming in on aspects like speaker impairment severity and phrase types.

Figure 3 shows the adapter performance drop δ per severity. For T-T, residual adapters seem to work similarly well across all 3 types of severities. On RNN-T models, we observe that δ is higher for speakers with moderate severity (median decrease of 18% for moderate vs 14% for mild and severe). In particular, we found that δ is somewhat correlated with the relative WER improvement γ of the fine-tuned model: δ is higher for cases where the adaptation by fine-tuning helps the most (Spearman correlation coefficient of 0.344 ($p < 0.001$) for RNN-T, more moderate correlation of 0.189 ($p=0.06$) for T-T). Overall, adaptation

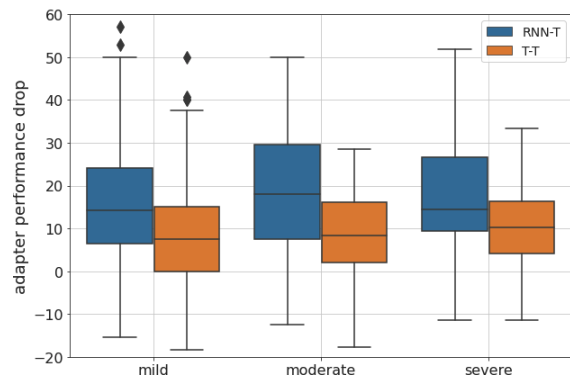


Figure 3: Distribution of adapter performance drop by severity of speech impairment.

Adaptation style	RNN-T			T-T		
	mild	moderate	severe	mild	moderate	severe
Unadapted	20.2	44.2	78.1	16.5	42.9	76.9
Fine-tune, full enc	4.1 (76%)	6.5 (84%)	14.2 (79%)	4.5 (74%)	6.9 (83%)	13.3 (78%)
Residual Adapters	4.8 (70%)	7.5 (80%)	16.3 (77%)	4.8 (71%)	8.2 (81%)	15.3 (77%)

Table 3: Results for atypical speech personalization task, broken down by severity (reported: median WER scores with relative WER improvement over the unadapted base model in brackets).

by fine-tuning and residual adapters show similar behavior across severity levels, which suggests that residual adapters do not have disadvantages for specific severity levels.

The Euphonia corpus also comes with domain information for each utterance. This enables us to analyze the performance of fine-tuning and residual adapters on two different domains – home automation queries⁵ (short phrases of 3.2 words on average) and conversational phrases (longer with 7.4 words on average, open domain) – to understand whether residual adapters have trouble with different phrase types. Table 4 shows T-T WERs for these two domains for a subset of 43 speakers who had a sufficient number of recordings of both home automation and conversational phrases. The conversational domain, being longer and with a more open vocabulary, generally is more challenging for ASR and accordingly across all severity levels we observe higher WERs. Moreover, this domain seems to be harder to adapt to, resulting in lower WER improvements through both types of adaptation, most notably on the severe group where adaptation gain drops from $\sim 85\%$ to $\sim 71\%$ (fine-tuning approach). Despite these difference, both fine-tuning and residual adapters show similar behavior, and we conclude that even for more challenging domains with longer utterances, residual adapters work well.

5.3 Accent Adaptation

Table 5 presents the per-accent WER results. Depending on the accent and the amount of training data, we observe substantial variance with respect to WER of the unadapted and adapted models. Across all accents, the Scottish accent (*sc*) performs worst with extremely high WER, both for the unadapted and adapted models.⁶ However, even for accents with fairly small amounts of training

⁵Examples: "turn on lights" or "play ABBA on Spotify".

⁶Note that beyond accent variation, results shown in Table 5 are affected by a domain mismatch between the training data used for the base model and the Common Voice corpus.

data, such as *a f*, adaptation clearly improves the performance over the unadapted model.

While fine-tuning the full encoder has slightly better performance than residual adapters, the adapter performance drop δ is relatively small (see Figure 2). This is consistent with our findings on the atypical speech personalization task (Section 5.2), where lower relative WER improvement is associated with much lower adapter performance drop. Analogously to the atypical speech personalization task, the adapter performance drop is smaller on T-T compared to RNN-T.

In order to provide a better context to the related work on accent recognition and to test the capability of the residual adapters in large group adaptation scenarios, we also ran experiments for multi-accent adaptation where a single model is fine-tuned (full encoder update) or adapted with residual adapters to all accents at once. We increased the bottleneck size to $b_d = 128$ so that the residual adapters have a sufficiently large capacity to handle this more complex task. A balanced training set of 11 accents (10 accents as described in Section 4.1 plus the US accent; 10k utterances per accent, up- or down- sampled depending on the amount of the training data per accent) was used for adaptation. Similarly, a balanced dev set was used to identify the best performing checkpoint. For WER calculation, we used the original test set per accent for comparability.

Results in Table 5 show that even in a large group adaptation scenario, residual adapters perform well and their performance is comparable to the per-accent adaptation scenario (adapter performance drop $\delta \approx 8\%$ for both RNN-T and T-T).

5.4 Training and inference time

On the atypical speech personalization task, when updating residual adapter parameters only – as opposed to encoder fine-tuning – we observed a speedup in training time as measured in global steps per second. Table 6 compares the median

Adaptation style	home automation			conversational		
	mild	moderate	severe	mild	moderate	severe
Unadapted	17.4	41.6	61.1	23.7	39.1	82.4
Fine-tune, full enc	3.7 (76%)	3.6 (88%)	7.8 (85%)	6.2 (67%)	4.9 (83%)	12.4 (71%)
Residual Adapters	4.7 (76%)	5.0 (87%)	8.5 (84%)	6.0 (65%)	5.8 (79%)	15.0 (72%)

Table 4: Results for atypical speech personalization task, broken down by domain (reported: median WER scores with relative WER improvement over the unadapted base model in brackets). Results are median scores over all speakers per severity group. We show results for T-T only due to space constraints; RNN-T results are analogous.

Accents		af	au	ca	en	in	ir	nz	ph	sc	si
# utt	train	1.6k	17.7k	13.2k	31k	20.1k	2.9k	2.3k	1k	4.4k	1.2k
	dev/test	300	2.2k	1.7k	3k	2.5k	360	300	300	550	300
RNN-T	Unadapted	16.1	17.3	11.5	13.7	20.0	11.6	13.4	18.3	56.3	20.5
	Per-Accent, Fine-tune	11.4	11.1	9.7	10.6	13.8	9.8	9.0	12.8	28.0	14.8
	Per-Accent, Res Adapt	11.8	12.3	10.3	11.0	15.3	10.1	10.1	13.9	30.9	15.6
	Multi-Acc, Fine-tune	11.7	12.6	10.4	11.6	15.5	9.8	9.4	12.3	30.6	15
	Multi-Acc, Res Adapt	12.2	13.3	10.5	12.1	16.9	10.4	10.9	14.1	33.6	16.3
T-T	Unadapted	16	18.9	13.2	15.3	20.8	13.7	13	19.9	61.2	24.2
	Per-Accent, Fine-tune	11.8	11.5	9.6	10.8	13.8	9.9	9.0	12.3	29.2	13.9
	Per-Accent, Res Adapt	12.5	12.0	9.8	11.1	14.7	9.9	10.3	13.5	31.9	15.3
	Multi-Acc, Fine-tune	11.2	12.8	9.7	11.3	15.2	9.5	9.5	11.9	33.5	14.3
	Multi-Acc, Res Adapt	12.7	14.1	10.3	11.7	16.2	10.7	10.1	13.1	36.1	14.8

Table 5: WER on the accent adaptation task, showing fine-tuning (full encoder) and residual adapters for the per-accent and multi-accent adaptation scenarios.

Arch	best checkpoint		global steps/sec	
	Fine-tune	Residual Adapters	Fine-tune	Residual Adapters
RNN-T	20380	5610	1.3	1.8
T-T	5150	5300	3.6	5.2

Table 6: Median number of steps for best checkpoint and global steps/second for the atypical speech personalization task showing that residual adapters lead to about 40% reduction in training time.

number of training steps needed across the 100 speakers (i.e. best checkpoint selected) as well as the global steps/second for fine-tuning the full encoder vs residual adapter training. While residual adapters led to about the same (T-T) or fewer (RNN-T) number of steps for convergence, the global steps/second score increased by about 40%.⁷ This gain is especially relevant for a personalization scenario where large numbers of user-specific models need to be trained.

During inference, on the other hand, we didn't observe a measurable increase in latency when

⁷We used the same accelerator setup (2x2 Tensor Processing Units slices) for fine-tuning and residual adapter training.

adding residual adapters. To test this, we decoded test sets of around 300 utterances several times on personalized models trained with and without residual adapters.

6 Conclusions and Future Work

In this work, we have shown that adaptation of ASR models using residual adapter layers leads to substantial WER improvements over unadapted models across two tasks: atypical speech (up to 77% relative WER reduction) and accented speech (up to 31% relative reduction) and in two architectures (RNN-T and T-T). In comparison, fine-tuning the entire encoder for each speaker or accent yields only small improvements compared to residual adapter training.

While similar in adaptation performance, residual adapters are much more parameter efficient than model fine-tuning. In our scenario, using residual adapters on each encoder layer, less than 0.5% of the overall model parameters need to be trained and maintained per speaker or accent. On the other hand, fine-tuning the entire encoder affects over 80% of the model parameters. In addition to substantially improved parameter efficiency, we also

observed a dramatic training time speed up of about 40% due to the reduced number of parameter updates.

Overall, these findings demonstrate a feasible and scalable solution for personalized, speaker-dependent models as well as domain-specific or dialect/accent-focused models.

In future work, we plan to study to which encoder layers we need to add adapters for best performance and to potentially make residual adapters even more parameter efficient. Similarly, we plan to apply residual adapters with different bottleneck dimensions depending on the position in the encoder layer stack (bottom and middle layers likely require larger, top layers smaller capacity). Finally, we also plan to directly compare the effectiveness of residual adapters to approaches using statically fed speaker-dependent vectors for speaker adaptation, especially in the context of accent adaptation.

References

- Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M. Tyers, and Gregor Weber. 2020. Common voice: A massively-multilingual speech corpus. In *Proc. LREC 2020*.
- Tom Bagby, Kanishka Rao, and Khe Chai Sim. 2018. Efficient implementation of recurrent neural network transducer in TensorFlow. In *IEEE Spoken Language Technology Workshop*, pages 506–512.
- Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proc. EMNLP-IJCNLP 2019*, pages 1538–1548.
- F. Biadsy, R. J. Weiss, P. J. Moreno, D. Kanvesky, and Y. Jia. 2019. Parrottron: An end-to-end speech-to-speech conversion model and its applications to hearing-impaired speech and speech separation. In *Proc. Interspeech 2019*, pages 4115–4119.
- Frederick L. Darley, Arnold E. Aronson, and Joe R. Brown. 1975. *Motor Speech Disorders*. Saunders.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proc. NAACL-HLT 2019*, pages 4171–4186.
- Rohan Doshi, Youzheng Chen, Jiang Liyang, Xia Zhang, Biadsy Fadi, Ramabhadran Bhuvana, Chu Fang, Andrew Rosenberg, and Pedro J. Moreno. 2020. Extending parrottron: An end-to-end, speech conversion and speech recognition model for atypical speech. In *Proc. ICASSP 2020*.
- Markus Freitag and Yaser Al-Onaizan. 2016. Fast domain adaptation for neural machine translation. *CoRR*, abs/1612.06897.
- Robert Gale, Liu Chen, Jill Dolata, Jan van Santen, and Meysam Asgari. 2019. Improving ASR Systems for Children with Autism and Language Impairment Using Domain-Focused DNN Transfer Techniques. In *Proc. Interspeech 2019*, pages 11–15.
- Shahram Ghorbani and John Hansen. 2018. Leveraging native language information for improved accented speech recognition. In *Proc. Interspeech 2018*, pages 2449–2453.
- Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Proc. ICASSP 2013*.
- Jordan E. Green, Robert L. MacDonald, Pan-Pan Jiang, Julie Cattiau, Rus Heywood, Richard Cave, Katie Seaver, Marilyn A. Ladewig, Jimmy Tobin, Michael P. Brenner, Philip C. Nelson, and Katrin Tomanek. 2021. Automatic speech recognition of disordered speech: Personalized models now outperforming human listeners on short phrases. In *Proc. Interspeech 2021*.
- Yanzhang He, Tara N Sainath, Rohit Prabhavalkar, Ian McGraw, Raziel Alvarez, Ding Zhao, David Rybach, Anjuli Kannan, Yonghui Wu, Ruoming Pang, et al. 2019. Streaming end-to-end speech recognition for mobile devices. In *Proc. ICASSP 2019*, pages 6381–6385.
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *Proc. ICML 2019*, pages 2790–2799.
- Anjuli Kannan, Arindrima Datta, Tara Sainath, Eugene Weinstein, Bhuvana Ramabhadran, Yonghui Wu, Ankur Bapna, and Zhifeng Chen. 2019. Large-scale multilingual speech recognition with a streaming end-to-end model. In *Proc. Interspeech 2019*.
- Robert L. MacDonald, Pan-Pan Jiang, Julie Cattiau, Rus Heywood, Richard Cave, Katie Seaver, Marilyn Ladewig, Jimmy Tobin, Michael P. Brenner, Philip Q. Nelson, Jordan R. Green, and Katrin Tomanek. 2021. Disordered speech data collection: Lessons learned at 1 million utterances from project euphonia. In *Proc. Interspeech 2021*.
- Meredith Moore, Hemanth Demakethapalli Venkateswara, and Sethuraman Panchanathan. 2018. Whistle-blowing asrs: Evaluating the need for more inclusive automatic speech recognition systems. *Proc. Interspeech 2018*, pages 466–470.
- Mumtaz Begum Mustafa, Siti Salwah Salim, Noraini Mohamed, Bassam Al-Qatab, and Chng Eng Siong. 2014. Severity-based adaptation with limited data for asr to aid dysarthric speakers. *PLOS ONE*, 9(1):1–11.

- Arun Narayanan, Rohit Prabhavalkar, Chung-Cheng Chiu, David Rybach, Tara N. Sainath, and Trevor Strohman. 2019. Recognizing long-form speech using streaming end-to-end models. In *IEEE Automatic Speech Recognition and Understanding Workshop*, pages 920–927.
- Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. 2019. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. In *Proc. Interspeech 2019*, pages 2613–2617.
- Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems*, volume 30.
- Tara N Sainath, Yanzhang He, Bo Li, Arun Narayanan, Ruoming Pang, Antoine Bruguier, Shuo-yiin Chang, Wei Li, Raziell Alvarez, Zhifeng Chen, et al. 2020. A streaming on-device end-to-end model surpassing server-side conventional model quality and latency. In *Proc. ICASSP 2020*, pages 6059–6063.
- George Saon, Hagen Soltau, David Nahamoo, and Michael Picheny. 2013. Speaker adaptation of neural network acoustic models using i-vectors. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 55–59.
- George Saon, Zoltan Tuske, Daniel Bolanos, and Brian Kingsbury. 2021. Advancing rnn transducer technology for speech recognition. In *Proc. ICASSP 2021*, pages 5654–5658.
- Joel Shor, Dotan Emanuel, Oran Lang, Omry Tuval, Michael Brenner, Julie Cattiau, Fernando Vieira, Maeve McNally, Taylor Charbonneau, Melissa Nollstadt, Avinatan Hassidim, and Yossi Matias. 2019. Personalizing ASR for Dysarthric and Accented Speech with Limited Data. In *Proc. Interspeech 2019*, pages 784–788.
- Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206.
- Pawel Swietojanski, Jinyu Li, and Steve Renals. 2016. Learning hidden unit contributions for unsupervised acoustic model adaptation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(8):1450–1463.
- Qian Zhang, Han Lu, Hasim Sak, Anshuman Tripathi, Erik McDermott, Stephen Koo, and Shankar Kumar. 2020. Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss. In *Proc. ICASSP 2020*, pages 7829–7833.
- Han Zhu, Li Wang, Pengyuan Zhang, and Yonghong Yan. 2019. Multi-Accent Adaptation Based on Gate Mechanism. In *Proc. Interspeech 2019*, pages 744–748.