Speech Personalization using Parameter Efficient Fine-Tuning for Nepali Speakers

Kiran Pantha[†], **Rupak Raj Ghimire**[†], and **Bal Krishna Bal**

Information and Language Processing Research Lab
Kathmandu University, Dhulikhel, Nepal
info@kiranpantha.com.np, rughimire@gmail.com, bal@ku.edu.np
[†]Equal contribution

Abstract

The performance of Automatic Speech Recognition (ASR) systems has improved significantly, driven by advancements in large-scale pre-trained models. However, adapting such models to low-resource languages such as Nepali is challenging due to the lack of labeled data and computational resources. Additionally, adapting the unique speech parameters of the speaker to a model is also a challenging task. Personalization helps to target the model to fit the particular speaker. This work investigates parameter-efficient fine-tuning (PEFT) methods like Low-Rank Adaptation (LoRA) and Decomposed Weight Low-Rank Adaptation (DoRA) to improve the performance of fine-tuned Whisper ASR models for Nepali ASR tasks by Personalization. These experiments demonstrate that the PEFT methods obtain competitive results while significantly reducing the number of trainable parameters compared to full fine-tuning. LoRA and DoRA show a relative WER to FT_{Base} increment of 34.93% and 36.79%, respectively, and a relative CER to FT_{Base} increment of 49.50% and 50.03%, respectively. Furthermore, the results highlight a 99.74% reduction in total training parameters.

1 Introduction

Automatic Speech Recognition (ASR) systems like voice assistants are widely used with the rapid development of deep learning models (Long et al., 2019). However, the system's performance depends on the diversity of the speech data. The model performs poorly on a different speaker with different speech characteristics that were not initially trained on. Personalization of the speaker helps fill that gap by making the ASR model work with the unique characteristics of the individual speaker. Training an ASR model requires high-quality speech data (Long et al., 2019; Radford et al., 2022). Collecting and training user-specific speech data for the model is challenging due to factors that consider user privacy. Most ASR applications are used on lightweight handheld devices with limited processing power. Customization of the model by fine-tuning the model based on user data is very difficult and inefficient due to the significant training parameters. Techniques such as residual adapter for fine-tuning (Tomanek et al., 2021), federated learning on devices by adopting the subset of weights (Jia et al., 2022a), and fine-tuning the model's attention and bias independently (Huang et al., 2021). ASR tasks for low-resource languages and Indo-Aryan languages like Nepali differ due to the language's structure and nature (Bal, 2004). Currently, Parameter Efficient fine-tuning (PEFT) is often used for Large Language Models (LLMs) because it lowers the computation power required to tune the model. The PEFT strategies, like LoRA and DoRA, are used to adapt the models with large trainable parameters. Due to the low use of resources by the adapted and merged model, it allows the large model to be inference using a consumer-grade GPU (Hu et al., 2021). Approaches like LoRA and DoRA are implemented for improving the efficiency of some state-of-the-art ASR models (Joseph and Baby, 2024; Yang et al., 2023). Here, an efficient speaker personalization approach using PEFT, two approaches, LoRA and DoRA, is proposed. The approach uses the low-rank approximation to efficiently adopt the ASR model for the targeted speaker with the limited weight addition, reducing the computation and memory restrictions (Hu et al., 2021; Liu et al., 2024).

2 Related Works

Traditional automatic speech recognition (ASR) architectures employed Hidden Markov Models (HMMs) together with Gaussian Mixture Models (GMM-HMM) to efficiently capture temporal dy-

Proceedings of the Fifth Workshop on Language Technology for Equality, Diversity, Inclusion, pages 189–198

namics and phonetic transitions (Rabiner, 1989). However, HMMs suffered in terms of speaker variability, strict temporal assumptions, and scalability limitations (Chakraborty and Talukdar, 2016). The development of hybrid architectures with Deep Neural Network - Hidden Markov Model (HMM-DNN) combinations improved robustness but at the expense of heavy computational resources (Li et al., 2013). Major architectures used for speech recognition are CNN (LeCun et al., 1989), LSTM (Hochreiter and Schmidhuber, 1997), BiLSTM (Schuster and Paliwal, 1997), RNN (Rumelhart et al., 1986), GRU (Chung et al., 2014). Traditional and recent ASR model uses a combination of the above architectures to perform speech transcription tasks in the Nepali ASR Domain (Regmi and Bal, 2021; Ghimire et al., 2024a). The transformer (Vaswani et al., 2017) based models work great for Nepali ASR tasks (Paudel et al., 2023). For speech personalization, different approaches like the Deep Neural Network (DNN) based acoustic modeling (Hinton et al., 2012), (Singular Value Decomposition) SVD based compression scheme (McGraw et al., 2016), user feedback (Mahesh Krishnamoorthy, 2016), controllable speech synthesis approach (Yang et al., 2023), enhancing quantized model (Zhao et al., 2023) were used. Before that, the data sparsity issue was solved by using speaker dependence with condensed vectors, reducing parameters during model adaptation using some regularization approach (Saon et al., 2013; Snyder et al., 2018; Fan et al., 2020; Sarı et al., 2020). The PEFT strategies used with LLMs includes methods like AdaptFormer (Chen et al., 2022), Visual Prompt Tuning (VPT) (Jia et al., 2022b), Low Rank Adaptation (LoRA) (Hu et al., 2021), Weight-Decomposed Low-Rank Adaptation (DoRA) (Liu et al., 2024), and Scaling & Shifting Your Features (SSF) ((Lian et al., 2022), among those the LoRA and the DoRA are implemented for improving the efficiency of the LLMs like GPT-2/3 (Radford et al., 2019), RoBERTa (Liu et al., 2019), and in some state-of-the-art ASR models (Joseph and Baby, 2024; Yang et al., 2023). Currently, the model is biased per speaker by fine-tuning the model's attention based on methods such as implementing a residual adapter for fine-tuning, federated learning on edge devices, and adopting a subset of weights. There is also a way of targeting a speaker for speech synthesis (Gabryś et al., 2022), which uses the cleaned version of speaker data for ASR Models to process (Wang et al., 2019). Speech data is preprocessed to improve speech parameters fed to the ASR model, thus enhancing the performance of the ASR model in noisy backgrounds (Wu et al., 2017). Experiments have also shown that a small volume of disordered speech of an individual's training data can benefit from Personalized ASR (Tobin and Tomanek, 2021). Fine-tuning the base model has also shown an improvement in the performance of the ASR for the Nepali language using active learning (Ghimire et al., 2023) and PEFT (Ghimire et al., 2024b). The Transformer (Vaswani et al., 2017) model has attention mechanism components, which are used by Low-rank adaptation as the Q, K, and V target modules for speech personalization, which can be used to personalize the ASR system without compromising the model's performance (Joseph and Baby, 2024). Many commercial products like Google Home[1], Amazon Alexa[2], Apple Siri[3], Microsoft Copilot[4], and different voice assistants also perform speech personalization to make interaction with users easier (Hoy, 2018).

The use of LoRA and DoRA is increasing in low-resource languages as well, and this motivated us to perform speaker personalization on the Nepali Language to improve the overall interaction with the ASR system.

3 Methodology

3.1 LoRA-based Speaker Personalization

LoRA (Low-Rank Adaptation)(Hu et al., 2021) technique is adopted to fine-tune the pre-trained weight matrices of the pre-trained Whisper (Radford et al., 2022) model as shown in Figure 2. Following LoRA (Hu et al., 2021), fine-tuning of a original pre-trained weight matrix W_0 (where $W_0 \in R^{d\times k}$; d and k are dimension of input feature vector and output feature vector respectively), the fine-tuning is limited by low-rank decomposition: $W' = W_0 + \Delta W = W_0 + BA$ where $B \in R^{d\times r}$ and $A \in R^{r\times k}$ with rank $r \ll \min(d, k)$. The pre-trained weight W_0 remains frozen, and only A and B are trainable, reducing the computational burden. A and B are multiplied by the same input, and their element-wise additions are calculated, and for an input vector $h = W_0x$, the new forward pass is $h = W'x = Wx + BAx$ (Hu et al., 2021).

[1]https://home.google.com
[2]https://alexa.amazon.com
[3]https://www.apple.com/siri/
[4]https://copilot.microsoft.com/

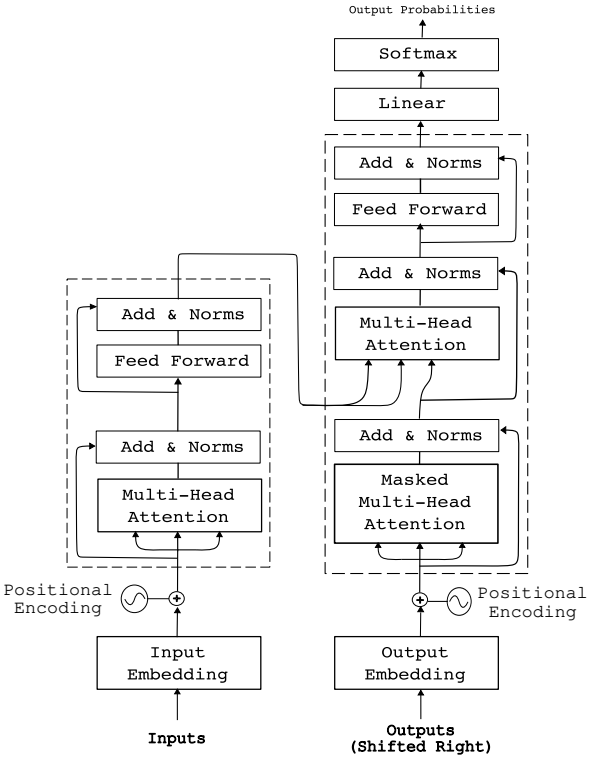

Figure 1: Architecture diagram of the Transformer model

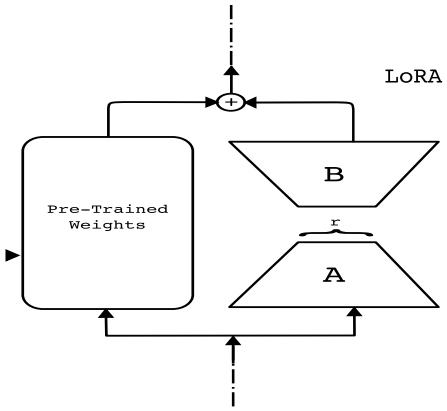

Figure 2: Architecture diagram of LoRA

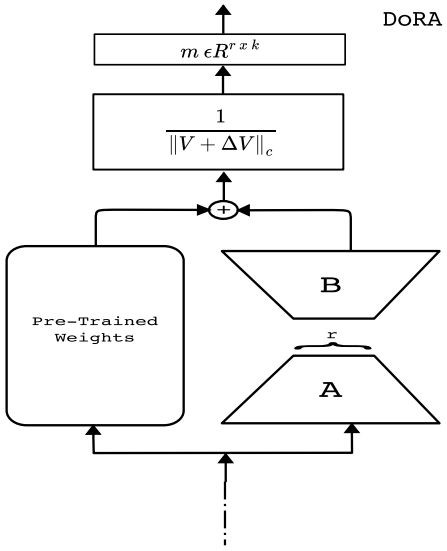

Figure 3: Architecture diagram of DoRA

3.2 DoRA-based Speaker Personalization

DoRA (Weight-Decomposed Low-Rank Adaptation) (Liu et al., 2024) technique is adopted where each weight matrix is mapped in each layer by incorporating another set of weight matrices, as outlined in Figure 3. As per DoRA (Liu et al., 2024), Weight decomposition is outlined as: $W_0 = m * (V/||V||_c) = ||W||_c * (W/||W||_c)$, where $m \in R^{1\times k}$ is the magnitude vector, $V \in R^{d\times k}$ is the directional matrix, and $||.||_c$ denotes the vector-wise norm across each column where d and k are dimension of input feature vector and output feature vector respectively. $W' = m * (V + \Delta V)/(||V + \Delta V||_c) = m * (W_0 + BA)/(||W_0 + BA||_c)$

Here, ΔV is the incremental directional update learned by the product of two low-rank matrices B and A. The matrices $B \in R^{d\times r}$ and $A \in R^{r\times k}$ are initialized according to DoRA's strategy so that W' is equal to W_0 before fine-tuning. DoRA enables more fine-grained updates across attention heads, improving adaptation efficiency (Liu et al., 2024).

3.3 Evaluation Metrics

In this research, Word Error Rate (WER), Character Error Rate (CER), Relative WER (RWER), and Relative CER (RCER) are used to evaluate the model's performance, rank selection, and target module combination selection in PEFT approaches. We discarded Match Error Rate (MER) from our evaluation as it is rarely used in ASR benchmarks and for script-specific nuances of the Nepali language in Devanagari script, where MER offers limited value over CER and complicates interpretations.

3.3.1 WER and CER

WER evaluates the accuracy of text recognition systems at the word level. CER evaluates similarly based on characters. Both metrics measure the percentage of incorrectly recognized words or characters, considering substitutions, insertions, and deletions.

$$\text{WER}\% = \frac{\text{Substitutions+Insertions+Deletions}}{\text{Total Words in Reference}} \times 100$$

Similarly, CER also evaluates the accuracy as WER, but at the character level.

3.3.2 Relative Metrices (RWER and RCER)

The Relative Word Error Rate (Relative WER) and the Relative Character Error Rate (Relative CER) are evaluative measures that examine the performance differences of a specified system compared to a reference baseline.

$$\text{Relative WER}\% = \frac{\text{WER}_{\text{system}} - \text{WER}_{\text{baseline}}}{\text{WER}_{\text{baseline}}} \times 100\%$$

$$\text{Relative CER}\% = \frac{\text{CER}_{\text{system}} - \text{CER}_{\text{baseline}}}{\text{CER}_{\text{baseline}}} \times 100\%$$

3.4 Methodology Details

Two variations of Low-Rank Adapters, namely, LoRA and DoRA, are proposed for speaker personalization to be used with the fine-tuned Whisper (Radford et al., 2022) a transformer (Vaswani et al., 2017) based model from OpenAI in the Huggingface (Wolf et al., 2019) ecosystem. The Proposed approaches for speaker personalization are hereby called **PEFT-LoRA** for the proposed model with LoRA and **PEFT-DoRA** for the proposed approach with DoRA. Another experiment is conducted to check the minimum amount of speech data required to PEFT fine-tune an ASR Model for the optimum rank found as per Table 2 for a set of speakers from Table 1.

Random Gaussian initialization is used to seed the trainable parameters for A; Zero initialization is used for B.ΔW $= BA$ is set at zero at the beginning of training so that the model can gradually learn to adapt to the Nepali language while retaining the pre-learned knowledge from the pre-trained Whisper model even with limited labeled data, making it a perfect fit for low-resource settings. Whereas DoRA / LoRA weights can be applied to any layer, our experiments focus on their integration into the query (W_q), key (W_k), and value (W_v) matrices of the attention mechanism which is in line with findings from previous research, which demonstrated how parameter-efficient approaches can be successfully used on these components to improve model performance (Radford et al., 2022; Hu et al., 2021; Liu et al., 2024; Huang et al., 2020).

Instead of fine-tuning all the parameters, LoRA and DoRA can be used to introduce low-rank learnable parameters update (ΔW) in attention layers, reducing computation cost while maintaining expressiveness (Radford et al., 2022; Liu et al., 2024; Hu et al., 2021).

4 Dataset

A portion of the CommonVoice17 (Ardila et al., 2020) dataset with slice/split of ne-NP/validated was taken for four speakers, and two speakers audio data was taken from the NepDS (Shishir Paudel and Bal Krishna Bal, 2022) dataset by taking the speaker having a commutative speech duration of more than 4 minutes. The data from both datasets is compiled and merged to form a compiled dataset (CommonVoice17 and ILPRL, 2025). The speaker-specific utterances are ordered reversely based on the number of utterances. First, six speakers were selected based on training data ranging from 4 to 18 minutes. Table 1 presents all of the properties of the speaker from the formed dataset labeled under the speaker column where the suffix CV means the speech data from CommonVoice17 and NEPDS means the speech data from the NepDS dataset along with speaker identification number, duration in minutes, number of utterances, and test-train split data. To identify the minimal amount of speech data for LoRA and DoRA PEFT implementation additional dataset with speech data as described in the Speech Range column of Table 4 is prepared.

Speaker	ID	Gender	Duration	Utterances	Train	Test
SpeakerNEPDS1	NS1	M	11.67	216	194	22
SpeakerNEPDS2	NS2	F	17.45	209	188	21
SpeakerCV1	S1	M	8.34	160	144	16
SpeakerCV2	S2	M	9.41	150	135	15
SpeakerCV3	S3	M	5.35	96	86	10
SpeakerCV4	S4	M	4.18	61	54	7

Table 1: Speaker Dataset for combined dataset

5 Experimental Setup

Transformer (Vaswani et al., 2017) Architecture is used for all our experiments implemented through the Huggingface architecture with PyTorch (Paszke et al., 2019) as the codebase. LoRA and DoRA weights are inserted into the Whisper Model Transformer Architecture for training. Every experiment is conducted on "Intel Data Center GPU Max 1550" GPU (Wu et al., 2024). The FT_{base} is a fine-tuned Whisper (Radford et al., 2022) model on OpenSLR54 (Kjartansson et al., 2018). For all the experiments, the Whisper Tokenizer that uses tiktoken (a byte pair tokenizer wrapper) decodes and encodes the dataset used for PEFT (Xu et al., 2023). The dataset (CommonVoice17 and ILPRL, 2025), model, and adapters from this research are

available in HuggingFace[5], and the code used is made available in GitHub[6].

5.1 Transformer Model Architecture

A basic block diagram of the Whisper Model (Transformer Model) is shown in Figure 1. This architecture contains an encoder-decoder structure where the encoder processes into audio features and generates a corresponding token, and the decoder decodes back the output text from the token predicted; the model leverages a multi-head attention mechanism and feed-forward layers to capture both local and global dependencies in the speech data. **Query (Q)**, **Key (K)**, and **Value (V)** are the metrics for self-attention in the transformer model, which PEFT later uses to adopt the Low-Rank Adaptation using LoRA and DoRA.

Layer Normalization and Residual Connections are also leveraged. The model has 1.55 billion parameters, with a 32-layer encoder and decoder, 16 attention heads, and a 1280-dimensional embedding space. It processes 80-dimensional Mel-spectrogram inputs, generates transcriptions using a vocabulary of 51,865 tokens, and employs rotary positional embeddings for efficient sequence modeling (Radford et al., 2022).

5.2 Experiment Details

Whisper Model (Radford et al., 2022) is fine-tuned with OpenSLR54 (Kjartansson et al., 2018) dataset to form a new fine-tuned base model for our experiment called FT_{Base} because the base model from Whisper (Radford et al., 2022) out-of-the-box performed poorly on the Nepali transcription task. The transformer-based encoder and decoder are modified for the Whisper model. Whisper has an encoder and decoder, each with multi-head self-attention and feed-forward networks. In the encoder, every self-attention layer is made up of weight matrices (W_q, W_k, W_v, and W_o) (Xu et al., 2022), initially 1280 × 1280 to match Whisper's embedding size. LoRA replaces these matrices with two smaller matrices, A and B, where A is of size 1280 × r and B is of size r × 1280. The rank of the matrix is obtained by attaching LoRA and DoRA components for ranks of 1 to 128. The final weight update is computed as $\Delta W = A \times B$, and this is added on top of the pre-trained weights. DoRA is also of a similar strategy but decouples rank from input-output dimensions so that updates can be applied separately per attention head. Since Whisper has 16 attention heads per layer, DoRA can better distribute updates across different model parts. LoRA and DoRA may be applied in the cross-attention layers of the decoder and even feed-forward networks, thus allowing fine-tuning without compromising the knowledge of the base model. The formula defines the number of new parameters introduced by LoRA: $LoRA_{params} = N \times m \times (Params\ of\ A, B) = N \times m \times (2 \times C \times r)$ and $DoRA_{params} = N \times m \times (Params\ of\ A, B, V) = N \times m \times (2 \times C \times r + C)$. Where N = Number of Encoder, m= Number of Matrices using DoRA or LoRA Weights, r = $LoRA/DoRA$ Rank, C = Encoder Cell Size (Hu et al., 2021; Liu et al., 2024) and training parameter for each rank is tabulated on Table 2, 3.

We use Rank (r) selection for the speaker adaptation, and further analyze the Query (Q), Key (K), and Value (V) Projection are used to best project the model performance based on the attention modules of the transformer model.

For Rank (r) selection, the Range of rank (r) values is evaluated to find the ideal value that fits with the speaker. The base Model is denoted by $Base$, fine-tuned models are denoted by FT. In the first case, the model is fine-tuned with the Nepali OpenSLR54 Dataset and is denoted as FT_{base}, which will act as a base model for our relative evaluation. In the second case, only the attention layer is fine-tuned (FT_A) as mentioned in Table 2 for every combination of target modules in the observed scenario (Huang et al., 2021). Based on the work by the authors of LoRA and DoRA, multiple sets of Query, Value, and Key Metrics of the attention layers (Hu et al., 2021; Liu et al., 2024) are selected. The subscript of FT_{Base} has the variant whether the FT_{base} is targeted on $C_{Attention}$ (c_attn), $Query(Q)$, $Key(K)$, and $Value(V)$. Experiments on rank are linked on the First Row of the Table. $FT_{A:kv}$ where key value metrics of the attention layer are being fine-tuned, and $FT_{A:qv}$ has query and value of the attention layer being fine-tuned. The number of Trainable Parameters is listed on the row labeled as TP. The ideal rank is determined by conducting several experiments as described above. For ranks above 64 and the training parameters are more significant than $FT_{A:qv}$; thus, ranks ranging from 1 to 32 are considered.

[5]https://hf.co/kiranpantha

[6]https://github.com/kiranpantha/LT-EDI-SPEECH

	Model	TP	WER%							AVG	RCER%	RWER %
			S1	S2	S3	S4	NS1	NS2	AVG	CER%		
	Base	-	88.34	89.93	89.36	82.07	86.54	87.17	87.24	30.17	-	-
	FT_{base}	1.554B	36.85	43.77	58.47	62.41	54.26	53.54	51.55	14.89	-	-
LoRA	$FT_{A:qv}$ $(r = 32)$	15.73M	38.14	40.38	43.16	52.24	17.93	17.26	34.85	8.25	44.59	32.40
LoRA	FT_{qv} $(r = 1)$	0.49M	41.24	40.38	83.16	50.75	20.69	25.22	43.57	8.84	40.62	15.47
LoRA	FT_{qv} $(r = 2)$	0.98M	42.27	32.69	40.00	56.72	20.69	14.16	34.42	8.22	44.80	33.23
LoRA	FT_{qv} $(r = 4)$	1.97M	36.08	34.62	48.42	49.25	21.38	19.47	34.87	8.69	41.64	32.36
LoRA	FT_{qv} $(r = 8)$	3.93M	37.11	40.38	45.30	47.76	16.55	14.16	33.54	7.52	**49.50***	**34.93***
LoRA	FT_{qv} $(r = 16)$	7.86M	43.30	40.38	43.16	52.24	18.62	19.03	36.12	8.55	42.58	29.93
LoRA	FT_{qv} $(r = 32)$	15.73M	38.14	40.38	43.16	53.73	20.69	16.37	35.41	8.42	43.45	31.31
DoRA	$FT_{A:qv}$ $(r = 32)$	15.97M	38.14	40.38	43.16	52.24	20.69	13.72	34.72	8.24	44.66	32.65
DoRA	FT_{qv} $(r = 1)$	0.737M	34.02	40.38	50.53	50.75	23.45	28.32	37.91	9.71	34.79	26.46
DoRA	FT_{qv} $(r = 2)$	1.228M	41.24	32.69	40.00	53.73	20.69	16.81	34.19	8.41	43.52	33.68
DoRA	FT_{qv} $(r = 4)$	2.211M	32.99	34.62	48.42	52.24	20.00	19.91	34.70	8.60	42.24	32.69
DoRA	FT_{qv} $(r = 8)$	3.932M	36.08	40.38	41.05	47.76	16.07	14.16	32.58	7.44	**50.03***	**36.79***
DoRA	FT_{qv} $(r = 16)$	8.110M	42.27	40.38	43.16	50.75	17.24	17.70	35.25	8.36	43.85	31.62
DoRA	FT_{qv} $(r = 32)$	15.97M	38.14	40.38	43.16	52.24	20.69	17.70	35.39	8.41	43.52	31.35

Table 2: Comparision of CER% and WER% accross different rank for LoRA and DoRA, * = selected row for rank based on highest RWER% and RCER%

	K	Q	V	TP	CER%							RCER%	WER%							RWER%
					S1	S2	S3	S4	NS1	NS2	AVG		S1	S2	S3	S4	NS1	NS2	AVG	
LoRA		✓		1.966M	10.04	11.34	11.26	11.62	4.33	7.93	9.42	36.74	45.36	44.23	45.26	52.24	21.38	22.57	38.51	25.30
LoRA	✓			1.966M	9.64	12.15	10.29	11.89	17.45	6.31	11.29	24.18	44.33	50.00	41.05	52.24	33.79	22.57	40.66	21.13
LoRA			✓	3.932M	9.44	9.31	10.1	10.27	3.84	5.50	8.08	45.74	38.14	38.46	46.32	50.75	20.00	18.58	35.38	31.37
LoRA	✓	✓		1.996M	10.44	9.31	10.68	10.00	4.21	6.88	8.59	42.31	47.42	28.85	44.21	50.75	20.69	26.99	36.48	29.23
LoRA		✓	✓	3.932M	10.64	9.72	9.32	11.08	4.21	3.72	8.12	45.47	42.27	40.38	40.00	49.25	22.07	14.16	34.69	32.71
LoRA	✓		✓	3.932M	7.23	10.53	8.54	11.08	3.71	7.28	8.06	**45.87***	28.87	38.46	42.11	52.24	20.00	22.57	34.04	**33.97***
LoRA	✓	✓	✓	5.898M	10.24	10.12	9.13	11.89	3.59	6.47	8.57	42.44	41.24	38.46	41.05	50.75	20.69	21.68	35.64	30.86
DoRA		✓		2.088M	8.84	11.74	10.68	11.62	4.33	8.01	9.20	38.21	41.24	46.15	42.11	52.24	20.69	23.01	37.57	27.12
DoRA	✓			2.088M	9.04	12.15	10.87	12.97	13.24	6.63	10.82	27.33	40.21	50.00	41.05	56.72	30.34	25.66	40.66	21.13
DoRA			✓	4.176M	9.04	9.72	10.29	11.62	3.71	5.58	8.33	44.06	37.11	42.31	45.26	49.25	19.31	19.03	35.38	31.37
DoRA	✓	✓		2.088M	10.44	9.31	10.49	10.27	3.84	8.90	8.87	40.43	45.36	28.85	44.21	52.24	20.69	31.42	37.13	27.97
DoRA		✓	✓	4.176M	10.84	9.72	8.93	10.54	4.46	3.80	8.05	45.94	42.27	40.38	40.00	47.76	22.07	14.60	34.51	33.06
DoRA	✓		✓	4.176M	6.83	10.93	8.54	10.54	3.71	7.20	7.96	**46.54***	31.96	38.46	42.11	47.76	21.38	22.57	34.04	**33.97***
DoRA	✓	✓	✓	6.264M	10.44	10.12	9.13	11.62	3.59	6.72	8.60	42.24	42.27	38.46	41.05	50.75	20.69	21.68	35.82	30.51

Table 3: Comparison of CER% and WER% across different attention layer configurations for LoRA and DoRA for Rank **(r) = 8**; * = selected row for Target Module combination based on highest RWER% and RCER%

For Query, Key, and Value projections, different target modules like $Query(Q)$, $Key(K)$, and $Value(V)$ are experimented to evaluate to analyze the model's performance. The rank is selected as per the initial rank analysis using the RCER and RWER values of the experiment for the given ranks. Each sub-module is taken for the experiment on each set of Q, K, and V parameters. The first column of the Attention Layer has three subdivisions of K, Q, and V, where ✓ means the model is activated on that group of target attention sub-modules. Moreover, the group of Attention Layer per speaker is tabulated below, where $S1$, $S2$, ..., $NS1$, $NS2$, TP, AVG, and $RCER$ have the same meaning as in the Table 2.

For Optimal Speaker Speech data, different quantities of speech data for a speaker were taken from a low of 1 minute to over 13 minutes, as in table 4. After that, the PEFT with the rank selected from optimal rank selection and query, key, and value are taken to apply the LoRA and the DoRA approaches to each commutative speech duration.

Duration (min)	Train Params	LoRA		DoRA	
		CER (%)	WER (%)	CER (%)	WER (%)
1	3.932M	13.71	49.62	13.74	50.31
2	3.932M	11.54	42.29	11.53	43.01
3	3.932M	11.04	42.50	11.04	42.83
4	3.932M	10.03	39.07	10.08	39.24
5	3.932M	9.70	35.84	9.44	37.37
6	3.932M	8.95	36.72	8.81	36.34
7	3.932M	8.81	35.49	8.86	35.38
8	3.932M	6.94	34.11	7.96	33.88
9	3.932M	4.91	22.83	4.93	22.96
10	3.932M	5.89	25.43	5.60	24.80
11	3.932M	6.21	23.20	6.16	22.94
12	3.932M	6.08	24.48	5.95	23.97

Table 4: Comparison of LoRA and DoRA for CER(%) and WER (%) metrics across cumulative speech duration ranges with Rank ($r = 8$), KV Target Model.

6 Results and Discussion

6.1 Rank (r) selection for proposed approach

Rank **8** is selected as the best good value for the rank (r) for personalization of a model, as the rank had the highest RCER and RWER from LoRA and DoRA approaches from Table 2 for the PEFT done on the ranks from 1 to 32 on QV target modules.

6.2 Query, Key and Value Projections

As per Table 3, the model's performance is excellent when the two-attention layers of **Key and Value** are taken for the selected rank of **r = 8**. The Key Value Pair has better RCER and RWER evaluation metrics than other attention modules. The **KV** combination performs better than the **KQ** combination. It gives better results, identical to the resulting pattern obtained from $FT_{A:kv}$, giving better results than $FT_{A:qv}$ (Huang et al., 2021).

6.3 Speech Data Duration for PEFT

The test results from table 4 reflected that just 1 minute of data yielded comparatively poor CER and WER metrics (around 13% CER). However, as more data was utilized, both these metrics continually improved. At around 3 minutes, CER dropped to around 10.5%, and at 5 minutes, to around 9%. Most importantly, when around 10 minutes of training data was used, CER kept dropping at around 5%, indicating good recognition performance.

7 Conclusion

Personalization of speech for the targeted speaker is quite challenging to fine-tune to fit the speech patterns. Using **CER** metrics implementing **LoRA** and **DoRA**, the finding shows **RCER** increment of **49.50%** and **50.03%** respectively. Similarly, for **WER** metrics implementing **LoRA** and **DoRA** shows **RWER** increment of **34.93%** and **36.79%** respectively. This result indicates that the DoRA approach performs better than the LoRA approach for both metrics. Further, the **K, V** combination of target module is found to be best performing in both LoRA and DoRA approaches using both metrics (RCER and RWER) as per Table 2. The result also shows a reduction of **99.74%** of total training parameters used to train and compute using PEFT compared to full fine-tuning.

Taking a system having CER < 5% to be our desired metrics, the findings suggest that around 10 minutes of speaker-dependent data is sufficient for effective fine-tuning with LoRA and DoRA, and it can serve as a good target for speech adaptation, especially personalized ASR applications in low-resource languages.

8 Limitations

Some limitations of the paper are highlighted in this section. Here, the personalization experiments were conducted on a small set of speakers (six in total, five Male and one Female), which may not sufficiently represent the full spectrum of Nepali language variation. Personalization was done in the Nepali language, so this is unclear how the LoRA/DoRA personalization approach will behave in other low-resource languages. Although showing improvement, it's unclear if the model would hold performance with speakers providing noisy, varied, or out-of-domain speech data.

References

R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber. 2020. Common Voice: A Massively-Multilingual Speech Corpus. In *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, pages 4211–4215.

Bal Krishna Bal. 2004. Structure of Nepali Grammer. In *PAN Localization Working Papers*, pages 332–396.

Chandralika Chakraborty and P.H. Talukdar. 2016. Issues and Limitations of HMM in Speech Processing: A Survey. *International Journal of Computer Applications*, 141:13–17.

Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. 2022. AdaptFormer: Adapting Vision Transformers for Scalable Visual Recognition. *arXiv preprint*. Version Number: 3.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv preprint*. Version Number: 1.

CommonVoice17 and KU ILPRL. 2025. Merged Dataset of CommonVoice17 and NepDS (ILPRL, KU) (Revision 3bfa307).

Zhiyun Fan, Jie Li, Shiyu Zhou, and Bo Xu. 2020. Speaker-aware speech-transformer. *arXiv preprint*. Version Number: 1.

Adam Gabryś, Goeric Huybrechts, Manuel Sam Ribeiro, Chung-Ming Chien, Julian Roth, Giulia Comini, Roberto Barra-Chicote, Bartek Perz, and Jaime Lorenzo-Trueba. 2022. Voice Filter: Few-shot text-to-speech speaker adaptation using voice conversion as a post-processing module. *arXiv preprint*. Version Number: 1.

Rupak Raj Ghimire, Bal Krishna Bal, and Prakash Poudyal. 2023. Active Learning Approach for Fine-Tuning Pre-Trained ASR Model for a Low-Resourced Language: A Case Study of Nepali. In *Proceedings of the 20th International Conference on Natural Language Processing (ICON)*, pages 82–89, Goa University, Goa, India. NLP Association of India (NLPAI).

Rupak Raj Ghimire, Bal Krishna Bal, and Prakash Poudyal. 2024a. A Comprehensive Study of the Current State-of-the-Art in Nepali Automatic Speech Recognition Systems. *arXiv preprint*. Version Number: 1.

Rupak Raj Ghimire, Prakash Poudyal, and Bal Krishna Bal. 2024b. Improving on the Limitations of the ASR Model in Low-Resourced Environments Using Parameter-Efficient Fine-Tuning.

Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Gerald Penn, and Sanjeev Khudanpur. 2012. Deep Neural Networks for Acoustic Modeling in Speech Recognition. In *ICASSP*, pages 2012–2016. IEEE.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Matthew B. Hoy. 2018. Alexa, Siri, Cortana, and More: An Introduction to Voice Assistants. *Medical Reference Services Quarterly*, 37(1):81–88.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv preprint*. Version Number: 2.

Yan Huang, Jinyu Li, Lei He, Wenning Wei, William Gale, and Yifan Gong. 2020. Rapid RNN-T Adaptation Using Personalized Speech Synthesis and Neural Language Generator. In *Interspeech 2020*, pages 1256–1260. ISCA.

Yan Huang, Guoli Ye, Jinyu Li, and Yifan Gong. 2021. Rapid Speaker Adaptation for Conformer Transducer: Attention and Bias Are All You Need. In *Interspeech 2021*, pages 1309–1313. ISCA.

Junteng Jia, Jay Mahadeokar, Weiyi Zheng, Yuan Shangguan, Ozlem Kalinli, and Frank Seide. 2022a. Federated Domain Adaptation for ASR with Full Self-Supervision. In *Interspeech 2022*, pages 536–540. ISCA.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. 2022b. Visual Prompt Tuning. *arXiv preprint*. Version Number: 2.

George Joseph and Arun Baby. 2024. Speaker Personalization for Automatic Speech Recognition using Weight-Decomposed Low-Rank Adaptation. In *Interspeech 2024*, pages 2875–2879. ISCA.

Oddur Kjartansson, Supheakmungkol Sarin, Knot Pipatsrisawat, Martin Jansche, and Linne Ha. 2018. Crowd-Sourced Speech Corpora for Javanese, Sundanese, Sinhala, Nepali, and Bangladeshi Bengali. In *6th Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU 2018)*, pages 52–55. ISCA.

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. 1989. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4):541–551.

Longfei Li, Yong Zhao, Dongmei Jiang, Yanning Zhang, Fengna Wang, Isabel Gonzalez, Enescu Valentin, and Hichem Sahli. 2013. Hybrid Deep Neural Network–Hidden Markov Model (DNN-HMM) Based Speech Emotion Recognition. In *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, pages 312–317.

Dongze Lian, Daquan Zhou, Jiashi Feng, and Xinchao Wang. 2022. Scaling & Shifting Your Features: A New Baseline for Efficient Model Tuning. *arXiv preprint*. Version Number: 3.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. 2024. DoRA: Weight-Decomposed Low-Rank Adaptation. *arXiv preprint*. Version Number: 6.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint*. Version Number: 1.

Yanhua Long, Yijie Li, Shuang Wei, Qiaozheng Zhang, and Chunxia Yang. 2019. Large-Scale Semi-Supervised Training in Deep Learning Acoustic Model for ASR. *IEEE Access*, 7:133615–133627.

Matthias Paulik Mahesh Krishnamoorthy. 2016. Improving Automatic Speech Recognition Based on User Feedback.

Ian McGraw, Rohit Prabhavalkar, Raziel Alvarez, Montse Gonzalez Arenas, Kanishka Rao, David Rybach, Ouais Alsharif, Hasim Sak, Alexander Gruenstein, Francoise Beaufays, and Carolina Parada. 2016. Personalized Speech recognition on mobile devices. *arXiv preprint*. Version Number: 2.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *arXiv preprint*. Version Number: 1.

Shishir Paudel, Bal Krishna Bal, and Dhiraj Shrestha. 2023. Large Vocabulary Continous Speech Recognition for Nepali Language using CNN and Transformer. In *Proceedings of the 4th Conference on Language, Data and Knowledge*, pages 328–333, Vienna, Austria. NOVA CLUNL, Portugal.

L.R. Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust Speech Recognition via Large-Scale Weak Supervision. *arXiv preprint*. Version Number: 1.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners.

Sunil Regmi and Bal Krishna Bal. 2021. An End-to-End Speech Recognition for the Nepali Language. In *Proceedings of the 18th International Conference on Natural Language Processing (ICON)*, pages 180–185, National Institute of Technology Silchar, Silchar, India. NLP Association of India (NLPAI).

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. 1986. Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*, pages 318–362. MIT Press, Cambridge, MA, USA.

George Saon, Hagen Soltau, David Nahamoo, and Michael Picheny. 2013. Speaker adaptation of neural network acoustic models using i-vectors. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 55–59, Olomouc, Czech Republic. IEEE.

Leda Sarı, Niko Moritz, Takaaki Hori, and Jonathan Le Roux. 2020. Unsupervised Speaker Adaptation using Attention-based Speaker Memory for End-to-End ASR. *arXiv preprint*. Version Number: 1.

M. Schuster and K.K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Shishir Paudel and Bal Krishna Bal. 2022. NepDS Dataset - Information and Language Processing Research Lab (ILPRL).

David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. 2018. X-Vectors: Robust DNN Embeddings for Speaker Recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5329–5333, Calgary, AB. IEEE.

Jimmy Tobin and Katrin Tomanek. 2021. Personalized Automatic Speech Recognition Trained on Small Disordered Speech Datasets. *arXiv preprint*. Version Number: 1.

Katrin Tomanek, Vicky Zayats, Dirk Padfield, Kara Vaillancourt, and Fadi Biadsy. 2021. Residual Adapters for Parameter-Efficient ASR Adaptation to Atypical and Accented Speech. *arXiv preprint*. Version Number: 1.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *arXiv preprint*. Version Number: 7.

Quan Wang, Hannah Muckenhirn, Kevin Wilson, Prashant Sridhar, Zelin Wu, John R. Hershey, Rif A. Saurous, Ron J. Weiss, Ye Jia, and Ignacio Lopez Moreno. 2019. VoiceFilter: Targeted Voice Separation by Speaker-Conditioned Spectrogram Masking. In *Interspeech 2019*, pages 2728–2732. ISCA.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv preprint*. Version Number: 5.

Bo Wu, Kehuang Li, Fengpei Ge, Zhen Huang, Minglei Yang, Sabato Marco Siniscalchi, and Chin-Hui Lee. 2017. An End-to-End Deep Learning Approach to Simultaneous Speech Dereverberation and Acoustic Modeling for Robust Speech Recognition. *IEEE Journal of Selected Topics in Signal Processing*, 11(8):1289–1300.

Hui Wu, Yi Gan, Feng Yuan, Jing Ma, Wei Zhu, Yutao Xu, Hong Zhu, Yuhua Zhu, Xiaoli Liu, Jinghui Gu, and Peng Zhao. 2024. Efficient LLM inference solution on Intel GPU. *arXiv preprint*. Version Number: 2.

Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. 2023. Parameter-Efficient Fine-Tuning Methods for Pretrained Language Models: A Critical Review and Assessment. *arXiv preprint*. Version Number: 1.

Sheng Xu, Yanjing Li, Teli Ma, Bohan Zeng, Baochang Zhang, Peng Gao, and Jinhu Lv. 2022. TerViT: An Efficient Ternary Vision Transformer. *arXiv preprint*. ArXiv:2201.08050 [cs].

Karren Yang, Ting-Yao Hu, Jen-Hao Rick Chang, Hema Swetha Koppula, and Oncel Tuzel. 2023. Text is All You Need: Personalizing ASR Models using Controllable Speech Synthesis. *arXiv preprint*. ArXiv:2303.14885 [eess].

Qiuming Zhao, Guangzhi Sun, Chao Zhang, Mingxing Xu, and Thomas Fang Zheng. 2023. Enhancing Quantised End-to-End ASR Models via Personalisation. *arXiv preprint*. ArXiv:2309.09136 [cs].