A memory-aware speaker adaptation method for disordered speech recognition

Yuhao Jiang
College of Software Engineering
East China Normal University
Shanghai, China
51255902128@ecnu.edu.cn

Weiting Chen*
College of Software Engineering
East China Normal University
Shanghai, China
wtchen@sei.ecnu.edu.cn

Jiahao Fan
College of Software Engineering
East China Normal University
Shanghai, China
51265902055@stu.ecnu.edu.cn

Abstract

Automatic speech recognition(ASR) is crucial for prompting effective communication in Patients with speech disorders. However, the high acoustic variability and data scarcity in disordered speech pose significant challenges to its recognition. Existing speech recognition methods struggle to address these challenges. Therefore, this paper proposes a memory-aware speaker adaptation method, enabling the ASR model to adapt to specific speakers while retaining common speaker features. Furthermore, this paper decouples speech representation into speaker information and semantic content. Accordingly, we develop two key modules: a memory-aware module, based on the read and write operations of a Neural Turing Machine, memorizes and retrieves speaker information; a content encoder, which includes a multi-level fusion module for aggregating semantic and acoustic features. Experimental results show that, compared to advanced speech recognition methods such as Transformer and time-delay neural networks(TDNN), our approach achieves 1.2% and 0.6% reduction in word error rate for aphasic speech and dementia speech, respectively.

CCS Concepts

• **Computing methodologies** → Artificial intelligence; Natural language processing; Speech recognition.

Keywords

disordered speech recognition, memory awareness, speaker adaptation

ACM Reference Format:
Yuhao Jiang, Weiting Chen, and Jiahao Fan. 2025. A memory-aware speaker adaptation method for disordered speech recognition. In *2025 6th International Conference on Computer Information and Big Data Applications (CIBDA 2025), March 14–16, 2025, Wuhan, China.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3746709.3746847

*Corresponding author.

CIBDA 2025, Wuhan, China

ACM ISBN 979-8-4007-1316-3/2025/03
https://doi.org/10.1145/3746709.3746847

1 Introduction

Severe speech disorders can result in language comprehension deficits in patients [1], increasing the likelihood of social rejection. Currently, approximately 5% of children have articulation disorders, such as stuttering, and up to 30% of stroke survivors experience aphasia [2]. At present, methods for diagnosing and treating speech disorders include speech and language therapy [3], acoustic parameter analysis [4], and neuropsychological scale assessments [5]. These methods put forward high requests for the expertise of speech pathologists, limiting treatment availability for many patients. Therefore, speech recognition technology can reduce labor costs and help alleviate the burden on physicians specializing in pathological speech.

There are two main challenges in speech recognition for disordered speech, compared to healthy speech. The first is high acoustic variability. Patients with disordered speech frequently manifest abnormal vocal patterns, distinct rhythms, and impaired articulation. The second is data scarcity. Existing datasets of disordered speech are generally small, and the corpus is often difficult to collect and of low quality. This limitation does not meet the data requirements of current ASR models, causing the models to overfit on the limited training samples.

To address these two challenges, current research [6] [7] commonly employs two approaches: speaker adaptation and self-supervised representation learning. The approach utilizing speaker adaptation achieves satisfactory results in reducing acoustic variability [7]. It combines speaker identity vectors [8], which provide information about speaker characteristics, such as articulation patterns, as prior knowledge to help ASR models adapt more effectively to inter-individual acoustic differences. The approach utilizing self-supervised learning (SSL) pretraining enables ASR models to utilize broader out-of-domain knowledge, helping to address data scarcity by extracting SSL representations and reconstructing input features [9].

Both speaker adaptation and self-supervised representation learning effectively enhance disordered speech recognition. However, since the speaker identity vector is extracted from the disordered speech dataset and the self-supervised representation from the healthy speech dataset, a domain mismatch arises, and an effective integration method for the two approaches is yet to be established. Therefore, in this paper, we propose a memory-aware speaker adaptation method that leverages memory capabilities to facilitate knowledge transfer across datasets and enhance the model's adaptability to different speakers. The contributions of this paper are summarized as follows:

- We combine speaker adaptation and self-supervised learning through memorization capabilities, a method not previously explored.
- We designed a novel partitioned memory pool using the neural Turing machine, enabling efficient memory access.
- Based on the partitioned memory pool, we designed a memory-aware module. The module stores speaker information over time and adapts to stylistic and acoustic variations across different speaker characteristics.
- We designed a content encoder featuring a multi-level fusion module, which interacts with the memory module for reading and writing.
- The experiment shows that our method has competitive performance compared with Transformer and time-delay neural networks(TDNN) methods.

2 Related Theory

This chapter outlines the methods and theories involved, as well as the baseline model used.

2.1 Speaker adaptation and SSL

Disordered speech exhibits significant acoustic variability, influenced by factors such as age, accent, and illness severity, often leading to substantial bias in training and test sets. To address acoustic variability, we apply the speaker adaptation technique that extracts auxiliary vectors **S** (e.g., i-vectors) representing the speaker's style and incorporates them into each frame of acoustic features **X**. This approach effectively introduces speaker prior knowledge during training, enhancing the model's generalization performance. The process can be expressed as equation(1), where **Y** represents the predicted text sequence S is the speaker identity vector, and f_θ is the ASR model.

$$Y = f_\theta (X + S) \tag{1}$$

Self-supervised representation learning (SSL) methods can greatly improve speech recognition. These models, trained on tens of thousands of hours of unlabeled speech without explicit supervision, predict masked portions of speech. They generate generic speech representations that enhance various downstream tasks, including speech recognition, speaker verification, and emotion recognition. Therefore, this paper integrates SSL representation to improve pathological speech recognition performance.

2.2 Neural turing machine

This paper builds memory perception capabilities based on a Neural Turing Machine (NTM) [26]. NTM is a memory augmented neural network, which consists of a controller, multiple independent read and write heads, and a memory pool, as illustrated in Figure 1. The controller receives external input h_t^l and generates a query vector q_t for each read and write head. The memory pool contains N memory vectors, which can be represented as $M = \{m_1, m_2, m_3, \ldots, m_N\} \in R^{D \times N}$.

When reading from memory, at each time step t, the read head takes the query vector q_t^r, generates the normalized weight vector

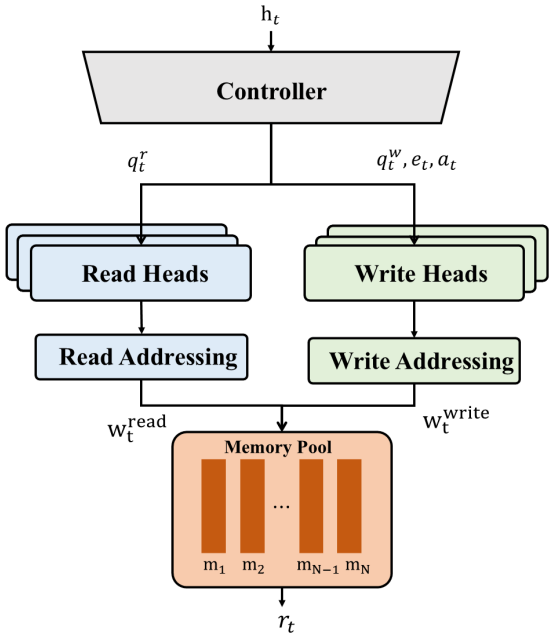

Figure 1: The Core Components of NTM.

w_t^{read}. And the resulting read output r_t is given by:

$$r_t = \sum_{i=1}^{N} w_t^{read}(i)\, m_i \tag{2}$$

When performing a memory write, the process is divided into two steps: memory erasure and addition. The write head generates the weight vector w_t^{read} *and* updates the memory using the erase vector e_t *and the add vector* a_t. *Each* memory location is updated according to the following equation:

$$m_i = m_i \left[1 - w_t^{write}(i)\, e_t(i)\right] + w_t^{write}(i)\, a_t(i) \tag{3}$$

2.3 Baseline model

We select E-branchformer [10] as the baseline model. It is based on the hybrid CTC/Attention architecture, with a structure similar to that of the Transformer, consisting of an encoder and a decoder. The encoder captures both global and local acoustic features. Its input is the hidden layer feature **X**, and the output is the captured acoustic feature **H**:

$$H = Enc(X) \tag{4}$$

E-branchformer contains two different decoders. The CTC decoder which uses conditional independence assumptions to predict the text of the current time frame, while the auto-regressive decoder is responsible for predicting the contextual text:

$$P_{CTC}(Y|X) = CTC(H) \tag{5}$$

$$P_{Dec}(y_\mu|X, y_{1:\mu-1}) = Dec\left(H, y_{1:\mu-1}\right) \tag{6}$$

where Y is the real labeled text sequence,mu is the currently decoded word. During training, the model is optimized by weighted combination of CTC loss and decoder loss.

$$\mathcal{L} = -\lambda \log P_{CTC}(Y|X) - (1-\lambda)\, P_{Dec}(Y|X) \tag{7}$$

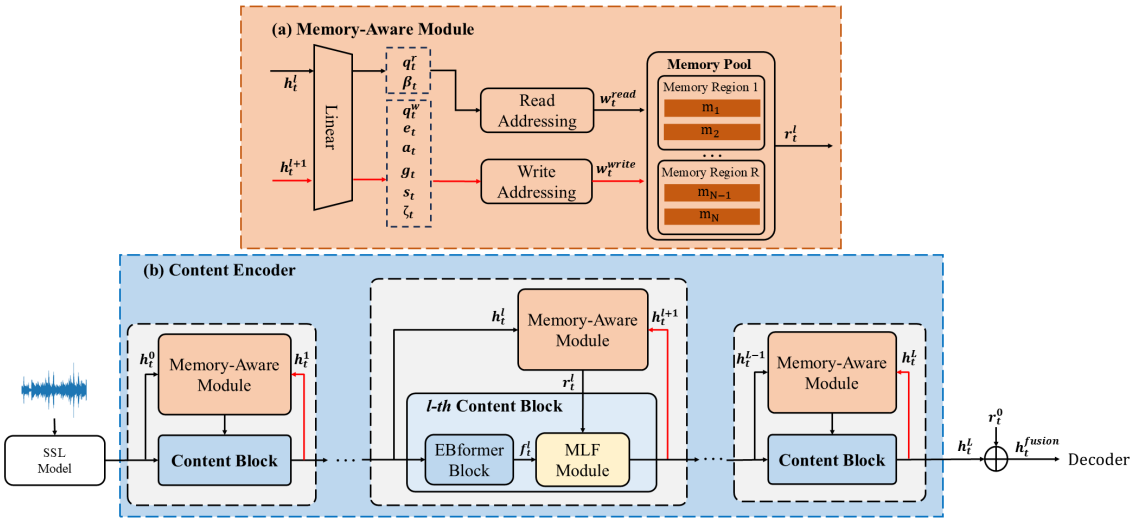

Figure 2: The encoder framework of the proposed memory-aware speaker adaptation method. Black solid arrows indicate data paths during both training and inference phases, while red solid arrows denote data paths exclusively in the training phase.

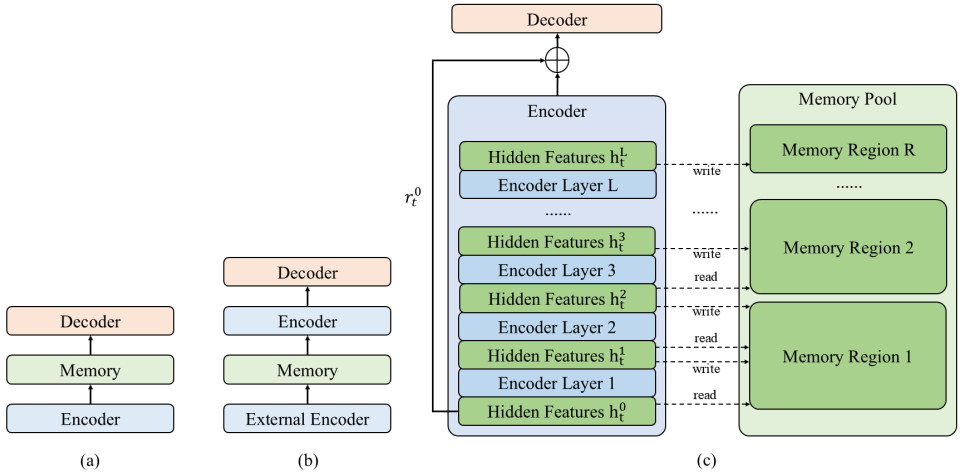

Figure 3: Comparison of different memory interaction strategies.

The weight λ is a hyper-parameter. The results of this model are usually evaluated using the word error rate (WER).

3 Methods

The architecture of the proposed method in this paper is shown in Figure 2, where Figure 2(a) is the memory-aware module and Figure 2(b) is the content encoder. We describe the design of the partitioned memory pool in Section 3.1, the structure of the memory-aware module and the memory interaction process in Section 3.2, and the structure of the content encoder and the multilayer fusion module in Section 3.3.

3.1 Partitioned Memory Pool

Recent studies [11-13] that utilize memory capacity have largely adopted the structures of Figure3(a) and Figure 3(b). We think that directly connecting memory to the encoder is not optimal, as various layers within the encoder network perceive speaker information differently. Therefore, We designed a partitioned memory pool with layer-by-layer memory reading and writing, which effectively facilitates the integration of speaker information with acoustic features, as is shown in the Figure 3(c).

For the hidden features h_t^0 in Layer 1, where information integration has not yet started, only memory reading is necessary. Accordingly, the hidden features h_t^L in the final layer L have completed information fusion, so we only update the memory pool. For

the general hidden features h_t^l , they serve both as the output of layer l, which is passed to the write head as a write query, and as the input to layer $l + 1$, which is sent to the read head as a read query.

We find that the weights within read-write weight vectors are not distributed evenly, typically clustering in a few specific dimensions. We think that each encoder layer focuses on different parts of the memory pool, often concentrating on only a few positions. Therefore, to achieve more precise and efficient addressing, we partition the memory pool into R non-overlapping regions, allowing each layer's read and write operations to access and update only the relevant memory region. we ensure that the read and write operations within the same layer address the same memory region. However, the memory vectors are not entirely independent and contain shared information present at a speaker level. When combined, this shared information can represent common speaker features, but memory partitioning may lead to incomplete common features. To alleviate this problem, we apply a residual connection of the read result r_t^0 from Layer 0 to the encoder's output.

Additionally, the time consumption of read and write operations mainly stems from calculating the read weight w_t^{read} , which involves N cosine similarity calculations. For L + 1 hidden layer features, a total of 2L read and write operations are needed, with each operation calculating the cosine similarity between the D-dimensional query vector and the N memory vectors. Without memory region partitioning, the overall time complexity is O(2NDL). After dividing the memory into regions, assuming each region contains P memory vectors, the overall time complexity reduces to O(2PDL). Since P is typically set to divide N integrally, which reduces the time complexity of cosine similarity computation, the division of memory regions helps to shorten the model's training time.

3.2 Memory-Aware Module

Based on the proposed partitioned memory pool, we design a memory-aware module for adaptive training. As shown in Figure 2(a), the module consists of a linear controller, several read/write heads, and a memory pool. In order to fully utilize the memory pool, it needs to be initialized first. We extract the frame-level speaker i-vectorsfrom the training data. We then average the frame-level i-vectors corresponding to the same sentence to obtain the sentence-level speaker features. Next, we utilize K-means clustering to reduce the number of speaker features by initializing the memory matrix with the N clustering centers as memory basis vectors. The embeddings obtained through clustering are more robust and effectively retain common speaker features during the highly aggregated computation process.

Then, we design a shared linear layer as the controller to generate the parameters $\{q_t^r, q_t^w, \beta_t, e_t, a_t, g_t, s_t, \zeta_t\}$, where the size of the linear layer matches the sum of the scalar and vector sizes involved for all read and write heads. The update of the linear layer's weight throughout the training process reflects the progressive establishment of long-term memory.

The read operation employs content-based addressing. At each time step t, the linear controller receives the hidden layer vector h_t^l and generates the read query vector q_t^r and modulation scalar β_t for each read head, as shown below:

$$\{q_t^r, \beta_t\} = Linear\left(h_t^l\right) \tag{8}$$

The read head then computes the cosine similarity between the query vector q_t^r and each memory vector m_i in the current memory region, generating the normalized weight vector w_t^{read} . β_t is a positive scalar that regulates the softmax result. A larger β_t leads to a steeper normalization curve, resulting in a more focused addressing range. The process for calculating the weight vector is shown below:

$$w_t^{read}(i) = softmax\left((\beta)_t Sim\left(q_t, m_i\right)\right) \tag{9}$$

where Sim denotes the computation of cosine similarity, $i \in \{KP, KP + 1, ..., (K + 1)P\}$, K represents the K − th memory region, and P represents the size of current memory region. The read operation only retrieves information from the memory pool without performing a ny updates. The outputs from each read head are concatenated to obtain r_t , which is then passed to a multi-level fusion module and added with $h_t^{l,fusion}$. The result of this addition becomes the input of the next layer h_t^{l+1} .

The write operation employs location-based addressing, allowing the write head to make fine-grained modifications to each memory location. At each time step t, the linear controller receives the output vector h_t^{l+1} from the current l−th layer and generates the following set of parameters for each write head:

$$\{q_t^r, e_t, a_t, g_t, s_t, \zeta_t\} = Linear\left(h_t^{l+1}\right) \tag{10}$$

where e_t and a_t represent the erase and add vectors, respectively. Additionally, to leverage the advantages of both addressing methods when updating the memory pool, an interpolation scalar $g_t \in (0, 1)$ is used to blend the current read weights w_t^{read} with the previous time step's write weights w_{t-1}^{read} . A higher g_t value leads the model to prefer content-based addressing, while a lower value makes it favor location-based addressing. When calculating the write weights for the first time, there are no previous time step write weights w_{t-1}^{read} available. In this case, we directly apply the erase and add operations to the read weights w_t^{read} .

$$w_t^g(i) = g_t w_t^{read}(i) + (1 - g_t) w_{t-1}^{write}(i) \tag{11}$$

To encourage the write head to concentrate on a localized range of weights, the linear controller generates a normalized shift weight vector s_t , addressing it within an allowable shift range. This procedure essentially combines adjacent elements with the shift weights, as shown below:

$$\bar{w}_t(i) = \sum_{j=1}^{P} w_t^g(j)\, s_t(i - j) \tag{12}$$

In practice, we implement shift weighting via circular convolution through circular convolution without bias. However, this convolution tends to diffuse the weights that were originally concentrated, potentially confusing the model's memory. To address this, the weight distribution is sharpened by ζ_t, resulting in the final write weights:

$$w_t^{write}(i) = \frac{(\bar{w}_t(i))^{\zeta_t}}{\sum_j^N (\bar{w}_t(j))^{\zeta_t}} \tag{13}$$

Notably, when the memory region size P is set too small, the read/write head inherently concentrates on weights in a localized area, making shift weighting unnecessary. In this case, we directly use the softmax result of w_t^g as w_t^{write} . Then, each write head independently updates the memory pool using w_t^{write} , the erase vector e_t , and the add vector a_t .

3.3 Content Encoder

To utilize speaker information to facilitate semantic content learning, we design a content encoder based on the E-branchformer , as illustrated in Figure 2(b). The Content Encoder consists of L sequentially connected Ebformer blocks (E-branchformer encoder blocks) and an MLF module. The EBformer block is used to learn content semantics, while the MLF block is responsible for feature aggregation. At the beginning and end of each content block, interactions with the memory-aware module are performed for read and write operations. During the write interaction, the content of the memory pool is updated, as indicated by the red data flow. However, the semantic content represents fine-grained feature-level information, whereas the read head provides coarse-grained discourse-level speaker information. Direct fusion of these two levels may result in a loss of detailed information.

To address information loss during multi-granularity feature fusion, we propose a multi-level fusion (MLF) module for aggregating features of varying granularities, as illustrated in Figure 4. The MLF module is mainly composed of the depthwise separable convolution (DSC) layer, pooling operations, and channel attention.

The DSC layer comprises three parallel branches, each performing depthwise convolution, ReLU activation, and pointwise convolution sequentially. These branches capture fine-grained features across multiple scales, producing aggregated features $\hat{f}_t^l$ through addition and batch normalization. Then, $\hat{f}_t^l$ is processed with mean pooling and maximum pooling in the time dimension to extract more compact information. The shared channel attention layer computes the weights w^{max} and w^{mean}, which are applied to $\hat{f}_t^l$ to generate the output h_t^{l+1}. The channel attention layer comprises two fully connected layer layers, along with ReLU and softmax activation functions. The process is as follows:

4 Experiement

To evaluate the effectiveness of the proposed method, we conduct comparative experiments and ablation studies on the AphasiaBank corpus. Additionally, we validate its generalizability using the DementiaBank corpus.

4.1 Datasets

AphasiaBank [14] is a commonly used public corpus in the study of aphasia. It consists of interview data from patients and clinicians. Most patients in AphasiaBank have stroke-induced aphasia and are classified into four severity levels based on aphasia quotient (AQ) scores [15]: mild (AQ > 75), moderate (50 < AQ ≤ 75), severe (25 < AQ ≤ 50), and v.severe (0 ≤ AQ ≤ 25). AphasiaBank contains conversation data from patients (PAR) and additionally collects data from non-aphasic individuals in the control group (INV). We follow the same data processing procedure and dataset division ratio as in [16], with 65.4 hours in the training set and 30.2 hours in the test

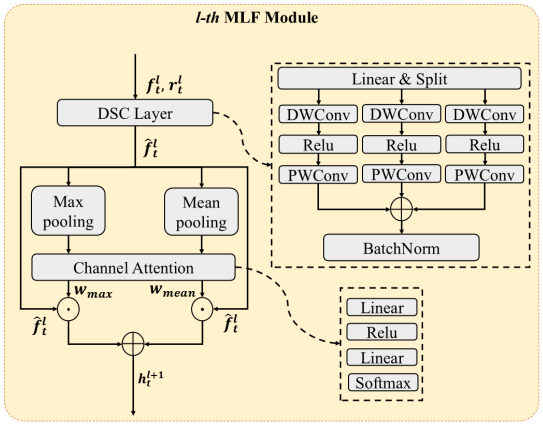

Figure 4: The structure of MLF module.

set. All experiments in this study use only the English subset of AphasiaBank and ensure no speaker overlap between the training, test, and validation sets.

DementiaBank [17] is a widely used corups in dementia research and consists of audio recordings from interviews, including picture descriptions, storytelling, and various tests. We use only the English subset, which includes audio recordings from 292 elderly participants and researchers. The DementiaBank english subset contains approximately 33 hours of data, with the validation and test sets comprising 2.5 hours and 0.6 hours, respectively.

4.2 Experimental setups

We use the ESPnet toolkit [22] to implement the models and experiments. The pre-trained SSL model used is wavLM-large [23]. The memory pool is configured with a fixed size of 256 rows and 64 columns, and it employs 2 read and write heads, respectively. The memory pool is segmented into 8 memory regions, each of size 8. In i-vectors extraction, the Universal Background Model (UBM) is trained with 1024 components, and the resulting i-vectors have a dimension of 256. The content encoder consists of 16 E-Branchformer layers, each with a hidden layer dimension of 256 and 4 attention heads. The cgMLP module has a hidden layer dimension of 3072, and the depthwise convolution uses a kernel size of 31. Furthermore, the convolutional kernels in the MLF module are set to sizes of 3, 15, and 31. The decoder is composed of 6 Transformer decoder layers, each with 2,048 hidden units and 4 attention heads. The weight ratio of CTC loss to AED loss is set to 3:7, with an inference beam size of 10. All experiments are conducted without external LM model-assisted decoding and conducted on a single NVIDIA 3090 GPU with 24 GB memory.

4.3 Results and discussion

Table 1 presents the Word Error Rate (WER) and Phonological Error Rate (PER) on AphasiaBank for all evaluated models. In general, due to the challenges of aphasic speech recognition, ASR models tend to have higher WERs, which increase as AQ improves. However, our proposed model consistently outperforms all the other models across all four severity levels. Specifically, our method achieved

Table 1: Comparison of PER or WER with various models on the AphasiaBank. "PAR" and "INV" refer to the patient and clinical investigator, respectively. Additionally, "Overall" represents the average WER for patients, while "ALL" denotes the average WER for both patients and clinical researchers.

Methods	Metric	PAR					INV	ALL
		Overall	Mild	Moderate	Severe	V.Severe		
DNN-HMM [2]	PER	-	47.4	52.8	61	75.8	-	-
DNN-HMM+MOE [18]	PER	36.9	33	41.6	62.9		-	-
BLSTM-RNN [19]	WER	-	33.7	41.1	49.2	63.2	-	-
Wav2vec2(zeroshot) [20]	WER	56	-	-	-	-	37.5	47.1
Wav2vec2 [9]	WER	-	23.6	36.8	36.4	59.1	-	-
WavLM-Transformer [21]	WER	-	22.8	33.4	35	56.6	-	-
Ebformer-InterCTC6 [16]	WER	25	21.1	28.4	29.6	40.6	16.1	20.2
ours	WER	**23.8**	**20.6**	**27.8**	**29.1**	**38.4**	**15.9**	**19.5**

Table 2: Comparison of WER with various models on the DementiaBank.

Methods	WER	
	PAR	ALL
Hybrid TDNN-SBE [24]	-	29.1
Ebformer-InterCTC9 [16]	31.2	28.9
Conformer [25]	29.7	27.6
Conformer-TDNN [24]	25.5	24.2
Ours	**24.6**	**23.6**

a mean (the Overall column) WER of 23.8% for the patient group (PAR) and WERs of 20.6%, 27.8%, 29.1%, and 38.4% at the mild, moderate, severe, and very severe levels, respectively Notably, in the very severe level of aphasia speech, characterized by fewer utterances and shorter average durations, our method achieves the greatest absolute reduction in WER compared to the second-best method, reaching a reduction of 2.2%. Although our work focuses on recognizing aphasic speech, it still achieves a 0.2% absolute reduction in the WER for the control group (INV), with a WER of 15.9%. The difference between the average WER of patients and the control group in our approach is 7.9%, which is 1% better than suboptimal, indicating that our model better adapts to the deviation of aphasic speech from healthy speech.

To validate the general applicability of our method, we evaluated it on the DementiaBank dataset. The results are presented in Table 2. Despite the generally poor audio quality of DementiaBank, with some vocalizations barely audible, our method, as shown in Table 2, still achieves best performance. It attains a mean WER of 24.6% on patient speech, representing a 0.9% absolute reduction compared to the sub-optimal method. In addition, our method also showed a 0.6% reduction in the total (the ALL column) WER compared to the suboptimal, confirming that our method is still effective on dementia speech.

In order to verify the effectiveness of the proposed modules, we display the results of the ablation study in Table 3. The baseline utilizes the E-branchformer, and the pre-trained SSL model adopts wavLM. The table shows that using SSL representations in place of traditional fbank features results in a total (the ALL column) WER of 20.2%, an absolute decrease of 13.2% compared to the 33.4% WER without SSL representations. This indicates that SSL representations captures more generalized speech feature information and effectively enhances the model's ability to handle disordered speech. When evaluating the influence of the memory-aware module, the total (the ALL column) WER reduces by 0.3% without SSL representations and 0.5% with SSL representations, compared to the baseline. This indicates that the speaker information retrieved by the memory-aware module can effectively assist the model in adapting to speech disorders of varying severity. Performance improves even further when combined with SSL, particularly with a 2.0% WER reduction at the extreme severity level. However, the memory-aware module only reduces the WER of INV by 0.1%, indicating that, regardless of SSLR use, the model's learning of healthy speech has become robust and stable. As a result, the memory read-write heads tend to output a domain-averaged speaker vector, exerting minimal influence on acoustic features. The multi-level fusion module achieves a 0.2% WER reduction for the INV group, as speaker characteristics typically manifest in subtle, localized changes, and our multi-level fusion design facilitates the integration of information at various scales. Results from the Prosody module indicate that introducing prosodic information slightly increases the WER at the mild severity level but significantly reduces WER at the moderate, severe, and very severe levels.

To explore the effect of the NTM read and write operations on speaker adaptation, We design an ablation study, as shown in Figure 5. Experiments from AphasiaBank indicate that the WER performance of the ASR model, enhanced by memory mechanism, consistently outperforms the baseline WER of 24.7% (with absolute reductions of 0.6% and 0.3%, respectively). This improvement is attributed to the model's ability to recall features that are most similar to those of the current speaker. However, the fixed content of the memory pool prevents knowledge transfer from the SSL model. So when both read and write operations are utilized, the WER declines by 0.9% and 0.7% compared to the baseline. This indicates that dynamically updating the memory pool during training better equips the model to handle complex and changing speakers, ultimately leading to the formation of a long-term and generalized memory. We observe a 0.3% decrease in WER with the layer-wise fine-grained

Table 3: Comparison of WER with various models on the DementiaBank. Ablation study on AphasiaBank, where "Mem" and "MLF" refer to the memory-aware module and multi-level fusion module.

SSL	Mem	MLF	PAR					INV	ALL
			Overall	Mild	Moderate	Severe	V.Severe		
w/o wavLM	×	×	35.7	32.1	38.8	39.5	49.1	31.0	33.4
	√	×	35.0	31.8	38.4	39.5	48.3	30.9	33.1
	√	√	35.0	31.7	38.5	39.5	48.1	30.7	32.9
w/ wavLM	×	×	25.0	21.1	28.4	29.6	40.6	16.2	20.2
	√	×	23.9	**20.4**	**27.8**	29.3	38.8	16.1	19.7
	√	√	**23.9**	20.5	28.1	**29.2**	**38.6**	**15.9**	**19.6**

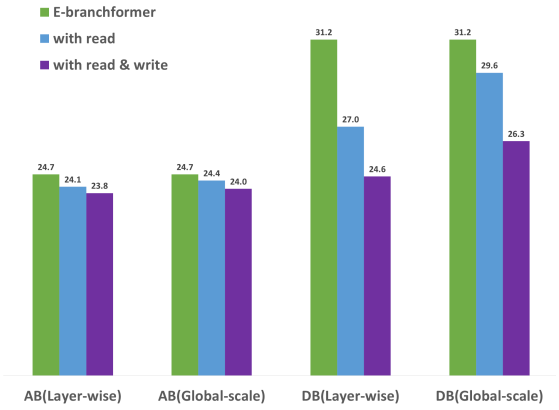

Figure 5: WER results for various interaction strategies on AphasiaBank and DentimaBank. "Layer-wise" fusion is our proposed memory read-write strategy, corresponding to Figure 2 (c), while global fusion corresponds to the approach shown in Figure 2 (a). "With Read" indicates the adoption of only the read mechanism, while "With Read & Write" indicates the adoption of both read and write operations.

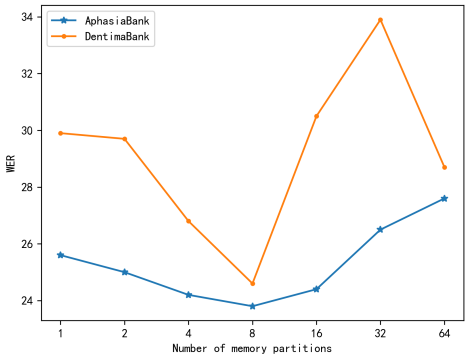

Figure 6: Effects of memory region size.

interaction approach compared to the global-scale approach, indicating that a hierarchical structure facilitates the retrieval and integration of memorized information. Similar trends are evident in the DentimaBank experiments, where our best results demonstrate a 6.6% reduction in overall WER compared to the baseline model.

To evaluate the impact of a memory pool with a total capacity of 64 on the model, we conducted an experiment varying the number of memory partitions. The results across two datasets are shown in Figure 6. Performance was poor when there was only a single region holding 64 memory vectors or 64 regions each holding one memory vector. The best results were achieved when the number of memory partitions was set to 8, with each memory region holding 8 memory vectors.

5 Conclusion

In this paper, we propose a memory-aware speaker adaptation method for disordered speech recognition. This method proposed a novel partitioned memory pool and leverages it for speaker adaptation. It enables the model to adapt to various speaker feature styles and acoustic offsets, effectively addressing the challenges of high speaker variability and data scarcity in disordered speech. Experimental results demonstrate that our method significantly outperforms existing approaches across multiple disordered speech datasets, with more pronounced improvements for patients with higher severity.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (Grant No. 61871186), and in part by the Dean's Fund of Engineering Research Center of Software/Hardware Co-design Technology and Application, Ministry of Education (East China Normal University).

References

[1] Grossman, M., 2010. Primary progressive aphasia: clinicopathological correlations. Nature Reviews Neurology 6, 88–97.
[2] Le, D., Provost, E.M., 2016. Improving automatic recognition of aphasic speech with aphasiabank, in: Interspeech, pp. 2681–2685.
[3] Monnelly, K., Marshall, J., Dipper, L., Cruice, M., 2023. Intensive and comprehensive aphasia therapy—a survey of the definitions, practices and views of speech and language therapists in the united kingdom. International Journal of Language & Communication Disorders 58, 2077–2102.
[4] Rehman, M.U., Shafique, A., Jamal, S.S., Gheraibia, Y., Usman, A.B., *et al.*, 2024. Voice disorder detection using machine learning algorithms: An application in speech and language pathology. Engineering Applications of Artificial Intelligence 133, 108047.
[5] Ciesielska, N., Sokołowski, R., Mazur, E., Podhorecka, M., Polak-Szabela, A., Kędziora-Kornatowska, K., 2016. Is the montreal cognitive assessment (moca) test better suited than the mini-mental state examination (mmse) in mild cognitive impairment (mci) detection among people aged over 60? meta-analysis. Psychiatr Pol 50, 1039–1052.

[6] Hu, S., Xie, X., Geng, M., Jin, Z., Deng, J., Li, G., Wang, Y., Cui, M., Wang, T., Meng, H., *et al.*, 2024. Self-supervised asr models and features for dysarthric and elderly speech recognition. IEEE/ACM Transactions on Audio, Speech, and Language Processing .

[7] Geng, M., Xie, X., Su, R., Yu, J., Jin, Z., Wang, T., Hu, S., Ye, Z., Meng, H.M., Liu, X., 2022. On-the-fly feature based rapid speaker adaptation for dysarthric and elderly speech recognition, in: Interspeech.

[8] Deng, J., Xie, X., Wang, T., Cui, M., Xue, B., Jin, Z., Li, G., Hu, S., Liu, X., 2023. Confidence score based speaker adaptation of conformer speech recognition systems. IEEE/ACM Transactions on Audio, Speech, and Language Processing 31, 1175–1190.

[9] Torre, I.G., Romero, M., Álvarez, A., 2021. Improving aphasic speech recognition by using novel semi-supervised learning methods on aphasiabank for english and spanish. Applied Sciences 11, 8872.

[10] Kim, K., Wu, F., Peng, Y., Pan, J., Sridhar, P., Han, K.J., Watanabe, S., 2023. E-branchformer: Branchformer with enhanced merging for speech recognition, in: 2022 IEEE Spoken Language Technology Workshop (SLT), IEEE. pp. 84–91.

[11] Carvalho, C., Abad, A., 2023. Memory-augmented conformer for improved end-to-end long-form asr, in: Interspeech, pp. 2218–2222.

[12] Sarı, L., Moritz, N., Hori, T., Le Roux, J., 2020. Unsupervised speaker adaptation using attention-based speaker memory for end-to-end asr, in: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE. pp. 7384–7388.

[13] Qiao, W., Bi, X., 2020. Ternary-task convolutional bidirectional neural turing machine for assessment of eeg-based cognitive workload. Biomedical Signal Processing and Control 57, 101745.

[14] MacWhinney, B., Fromm, D., Forbes, M., Holland, A., 2011. Aphasiabank: Methods for studying discourse. Aphasiology 25, 1286–1307.

[15] Kertesz, A., 2006. Western aphasia battery-revised (wab-r). Austin, TX: Pro-Ed 10.

[16] Tang, J., Chen, W., Chang, X., Watanabe, S., MacWhinney, B., 2023. A new benchmark of aphasia speech recognition and detection based on e-branchformer and multi-task learning, in: Interspeech.

[17] Becker, J.T., Boiler, F., Lopez, O.L., Saxton, J., McGonigle, K.L., 1994. The natural history of alzheimer's disease: description of study cohort and accuracy of diagnosis. Archives of neurology 51, 585–594.

[18] Perez, M., Aldeneh, Z., Provost, E.M., 2020. Aphasic speech recognition using a mixture of speech intelligibility experts, in: Interspeech.

[19] Le, D., Licata, K., Provost, E.M., 2018. Automatic quantitative analysis of spontaneous aphasic speech. Speech Communication 100, 1–12.

[20] Chatzoudis, G., Plitsis, M., Stamouli, S., Dimou, A.L., Katsamanis, N., Katsouros, V., 2022. Zero-shot cross-lingual aphasia detection using automatic speech recognition, in: Interspeech, pp. 2178–2182.

[21] Perez, M., Le, D., Romana, A., Jones, E., Licata, K., Provost, E.M., 2023. Seq2seq for automatic paraphasia detection in aphasic speech. arXiv preprint arXiv:2312.10518 .

[22] Mehta, S., Rastegari, M., Caspi, A., Shapiro, L., Hajishirzi, H., 2018. Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation, in: Proceedings of the european conference on computer vision (ECCV), pp. 552–568.

[23] Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S., Chen, Z., Li, J., Kanda, N., Yoshioka, T., Xiao, X., *et al.*, 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. IEEE Journal of Selected Topics in Signal Processing 16, 1505–1518.

[24] Geng, M., Xie, X., Ye, Z., Wang, T., Li, G., Hu, S., Liu, X., Meng, H., 2022. Speaker adaptation using spectro-temporal deep features for dysarthric and elderly speech recognition. IEEE/ACM Transactions on Audio, Speech, and Language Processing 30, 2597–2611.

[25] Hu, S., Xie, X., Jin, Z., Geng, M., Wang, Y., Cui, M., Deng, J., Liu, X., Meng, H., 2023. Exploring self-supervised pre-trained asr models for dysarthric and elderly speech recognition, in: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE. pp. 1–5.

[26] Graves, A., 2014. Neural turing machines. arXiv preprint arXiv:1410.5401 .