frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 12 December 2024
DOI 10.3389/fphy.2024.1404503

OPEN ACCESS

EDITED BY
Gongye Zhang,
Southeast University, China

REVIEWED BY
Yassine Himeur,
University of Dubai, United Arab Emirates
Peter Pocta,
University of Žilina, Slovakia

*CORRESPONDENCE
Cuimei Liu,
tracyliu20132024@163.com

RECEIVED 21 March 2024
ACCEPTED 25 November 2024
PUBLISHED 12 December 2024

CITATION
Wu Y, Luo X, Guo F, Lu T and Liu C (2024) Research on multi-scenario adaptive acoustic encoders based on neural architecture search. *Front. Phys.* 12:1404503.
doi: 10.3389/fphy.2024.1404503

Research on multi-scenario adaptive acoustic encoders based on neural architecture search

Yiliang Wu[1,2], Xuliang Luo[1,2], Fengchan Guo[1], Tinghui Lu[1,2] and Cuimei Liu[1]*

[1]Guangdong Power Grid Co., Ltd., Jiangmen Power Supply Bureau, Jiangmen, China, [2]Faculty of Intelligent Manufacturing, Wuyi University, Jiangmen, China

This paper presents the Scene Adaptive Acoustic Encoder (SAAE) method, which is tailored to diverse acoustic environments for adaptive design. Hand-crafted acoustic encoders often struggle to adapt to varying acoustic conditions, resulting in performance degradation in end-to-end speech recognition tasks. To address this challenge, the proposed SAAE method learns the differences in acoustic features across different environments and accordingly designs suitable acoustic encoders. By incorporating neural architecture search technology, the effectiveness of the encoder design is enhanced, leading to improved speech recognition performance. Experimental evaluations on three commonly used Mandarin and English datasets (Aishell-1, HKUST, and SWBD) demonstrate the effectiveness of the proposed method. The SAAE method achieves an average error rate reduction of more than 5% compared with existing acoustic encoders, highlighting its capability to deeply analyze speech features in specific scenarios and design high-performance acoustic encoders in a targeted manner.

KEYWORDS

automatic speech recognition, acoustic encoder, acoustic features, neural architecture search, multi-scenario

1 Introduction

The rapid evolution of human-computer intelligent interaction recently has propelled automatic speech recognition (ASR) [1] to the forefront as a crucial technology for intelligent interactive applications. ASR has extensive utility across various application scenarios, such as voice search [2], voice assistants [3], meeting minutes [4], intelligent services [5], and robots [6]. Over the past few decades, with the rapid advancement of computer technology and machine learning, speech recognition technology has made significant progress. Currently, speech recognition systems can recognize continuous speech [7], handle various accents and dialects [8], and can be applied to various practical applications.

Speech recognition technology plays a significant role in people's daily work and lives. It facilitates communication and operation for individuals with visual or motor impairments, enabling them to interact with others or operate machines [9]. In the automobile industry, speech recognition technology enables the hands-free operation of in-vehicle entertainment systems, contributing to driver safety and convenience [10]. Within office environments, speech recognition expedites the conversion of spoken words into text, thereby enhancing

productivity [11]. In customer service, automated voice response services [12] reduce the need for manual customer service. Within healthcare, speech recognition aids healthcare professionals in medical record maintenance and transcription, saving time and enhancing accuracy [13]. Furthermore, in education, speech recognition serves as a tool for evaluating students' language learning progress [14]. In essence, speech recognition technology not only enriches daily routines but also continually refines our quality of life.

Among the existing speech recognition technologies, end-to-end ASR stands out as the foremost approach [15–18]. Within end-to-end ASR systems, the acoustic encoder is of vital significance. In practical applications, ASR systems encounter diverse scenarios, each presenting different acoustic characteristics influenced by factors such as pronunciation, speaking style, and emotional tone. These differences emanate from both speakers and speech content, involving various forms such as telephone conversations, scripted speech, and in-vehicle interactions. Moreover, linguistic disparities across languages such as Mandarin Chinese, Cantonese, and English introduce distinct pronunciation rules, further contributing to different acoustic characteristics [19, 20]. In addition, real-world speech data often contend with environmental noise and reverberation, leading to significant variations in acoustic properties even when speech content and language remain constant. Navigating these challenges to enhance the acoustic encoder's ability to accurately model speech features under specific scenarios represents a formidable task.

To obtain an acoustic encoder adaptable to specific acoustic scenarios, a common approach is to employ an existing acoustic encoder structure and train it with speech data from the target scenario, thus achieving effective modeling. In previous literature [21], an acoustic encoder based on the recurrent neural network (RNN) structure was employed, using temporal modeling of speech data and training within the RNN-transducer (RNN-T) framework. Another study [22] introduced an acoustic encoder with a more efficient multi-head attention mechanism, which is capable of globally and in parallel modeling the speech sequence, thereby enhancing the encoder's modeling capability. This encoder is effectively trained using the connectionist temporal classification (CTC) framework [23, 24]. Similarly, in another study [25], an acoustic encoder based on the multi-head attention mechanism was employed within an attention-based encoder-decoder framework (AED). The AED framework can use existing text information to assist the acoustic encoder in modeling acoustic features in speech, further refining its modeling capability. Moreover, within RNN-T frameworks such as CTC and AED, studies have explored the use of acoustic encoders based on convolutional neural networks (CNNs) or their variants for ASR tasks.

However, acoustic encoders with fixed structures encounter challenges in adapting to diverse acoustic scenarios. To address this challenge, manual adjustment of the encoder structure is a common approach. Nonetheless, this approach presents two significant problems: First, the number of manually designed encoder structures is limited, making it uncertain whether the most suitable encoder for the target scenario can be devised. Second, verifying the performance of manually designed encoders entails training and evaluating each encoder separately, resulting in high design costs for practical applications.

To address the challenge inherent in manual encoder design, this paper proposes a method for designing Scene Adaptive Acoustic Encoders (SAAE) using neural architecture search [26–28]. This approach addresses the limitations of manual designs using a two-pronged strategy. First, a novel search space tailored to the requirements of acoustic encoders for end-to-end ASR tasks is developed. This search space comprehensively considers the acoustic characteristics of speech data across different scenarios, offering a range of candidate encoder structures suitable for modeling various acoustic features. Second, the differentiable architecture search algorithm DARTS is employed to identify the optimal encoder structure for the target scenario. To further enhance performance, the Gumbel re-sampling technique and a corresponding pre-training search strategy are employed. Experiment validation was conducted on three Mandarin and English datasets under different scenarios. Results show that SAAE effectively reduces error rates compared with various baseline encoders with different structures, affirming its capability to design high-performance acoustic encoders tailored to specific acoustic scenarios.

2 Scene adaptive acoustic encoder structure design

The scene Adaptive Acoustic Encoder Structure Design (SAAE) method begins by devising a novel search space based on the domain-specific knowledge of ASR tasks. This search space provides abundant candidate encoder structures tailored for speech recognition tasks. Subsequently, SAAE introduces and refines a differentiable search algorithm to discern the appropriate encoder structure from the search space based on the target scene. Once the encoder structure is identified, SAAE proceeds to retrain the encoder to accomplish the final recognition task.

2.1 Encoder search space

The three predominant end-to-end ASR frameworks, namely, CTC, AED, and RNN-T, serve as the foundation for defining each encoder search. Figure 1 shows the specific structures of these three frameworks. In addition to the feature preprocessing module required by the end-to-end ASR system, all three frameworks need an acoustic encoder to extract high-dimensional characterization information h^{enc} from the input speech features $\{x_1,\cdots,x_t\}$ at a time t, and then send it to their respective downstream units to derive the prediction of the u-th output text $\hat{y}_u$ and further complete the recognition decoding task. The CTC framework uses a CTC classifier to process h^{enc} and predict the output label using the softmax function. In contrast, the AED framework uses an autoregressive decoder [29] to receive h^{enc} and the historical text $\{y_0,\cdots,y_{u-1}\}$ simultaneously to predict the current output; RNN-T uses a dedicated prediction network to process $\{y_0,\cdots,y_{u-1}\}$ and obtain the text prediction information P_u of the current u-th output, and finally gives the prediction of the current output by combining the information in h^{enc} and P_u.

Unlike existing strategies for manually designing encoder structures, SAAE specifically designs a search space for

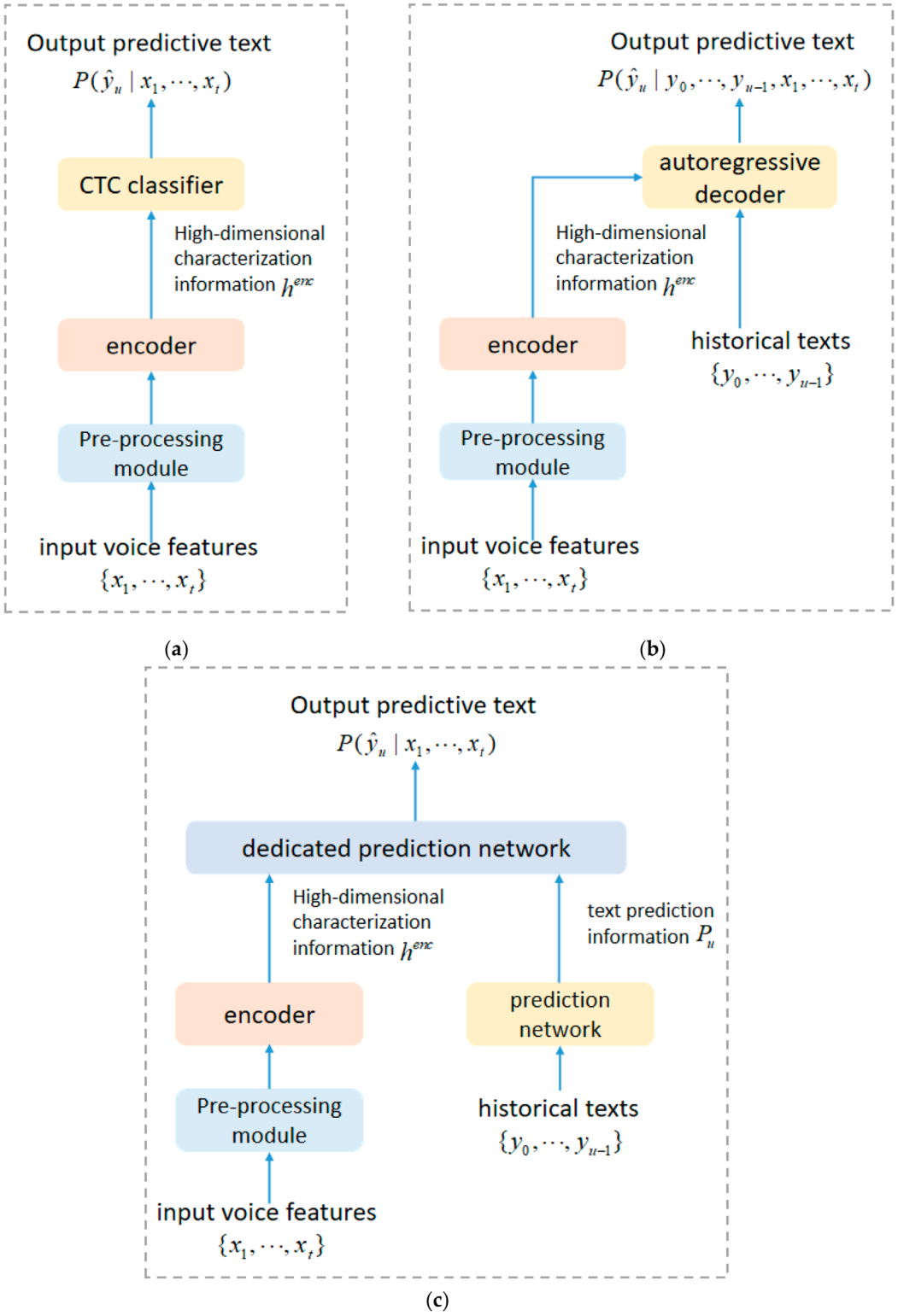

FIGURE 1
Three end-to-end speech recognition frameworks: **(A)** CTC framework, **(B)** AED framework, and **(C)** RNN-T framework.

the encoder in the end-to-end ASR framework. Figure 2 shows a schematic diagram of the entire SAAE encoder search space.

As shown in Figure 2A, the SAAE encoder comprises N layers of searchable SAAE modules, with each SAAE module offering a diverse array of candidate operations. Through a systematic

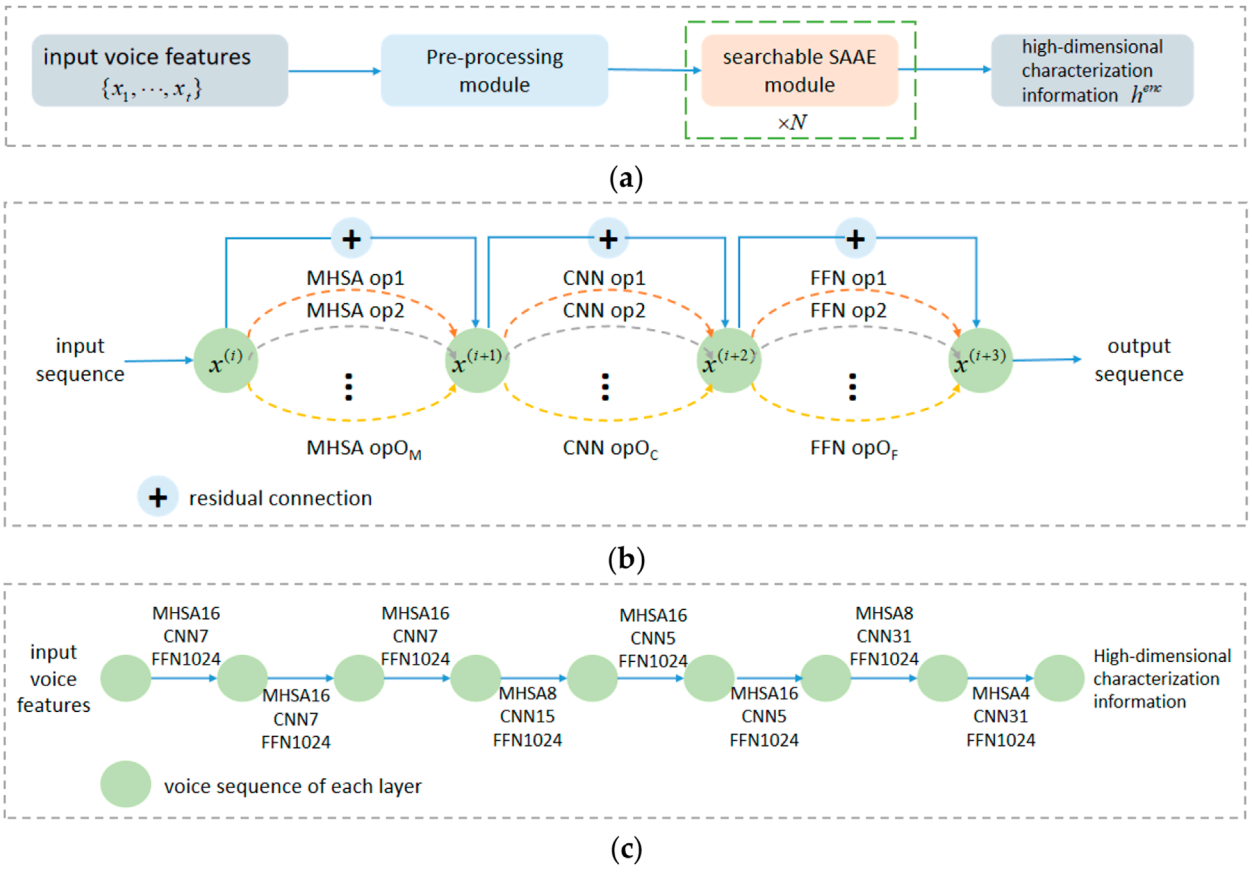

FIGURE 2
SAAE encoder search space: **(A)** the overall framework, **(B)** specific results of the SAAE module, and **(C)** an example of a search on the Aishell-1 dataset.

search and combination of SAAE modules layer by layer, the entire search space furnishes an extensive range of candidate encoder structures tailored to accommodate various scenarios. Within each searchable module, SAAE meticulously considers the characteristics inherent to the ASR task, incorporating multi-head self-attention modules (MHSA), CNNs, and feedforward neural network modules (FFN) with distinct functionalities to construct the acoustic encoder. Figure 2B illustrates the composition of a module that integrates three distinct functional modules.

The MHSA module employs a unique self-attention mechanism to process the global context of the input series, facilitating the extraction of deep acoustic features [30]. Meanwhile, the CNN module employs convolutional operations to model the local intercorrelation of acoustic signals, thereby uncovering latent information in speech signals [31]. Finally, the FFN module is tasked with mapping the current signal sequence to a higher-dimensional hidden layer for nonlinear activation, thereby enhancing the model's nonlinear fitting capability [32]. Figure 2B demonstrates the specific configuration of the searchable SAAE module, which comprises the above three different functional modules. The overall structure is abstracted into a directed acyclic graph, where every node $x^{(i)}$ signifies the output of the preceding module $x^{(i-1)}$, and the lines between adjacent nodes represent the respective candidate operations offered by the current functional module. Additionally, residual connections are introduced into the SAAE module, which has been previously proven effective in improving the generalization performance of the model [33].

Based on three different functional modules, SAAE provides various candidate operations according to the structural characteristics of the group. In Figure 2B, O_M, O_C, and O_F represent the number of candidate operations corresponding to each module. SAAE provides a flexible setting for the number of attention heads in the candidate operations of MHSA, such as 4, 8, and 16, which are denoted as MHSA4, MHSA8, and MHSA16, respectively. For the CNN module, SAAE provides multiple candidate operations with different kernel sizes to process acoustic features of different ranges, such as kernels 7, 15, and 31, which are denoted as CNN7, CNN15, and CNN31, respectively, and an additional Skip candidate operation. For the FFN module, SAAE offers different dimensions of hidden layers, such as 256, 512, and 1,024, which are denoted as FFN256, FFN512, and FFN1024, respectively, to explore the influence of the FFN hidden layer dimension on the performance of the acoustic encoder. In this study, we set the number of layers N of the SAAE module to 8, so that the number of possible candidate encoders in the entire search space can be obtained $(3 \times 4 \times 3)^8 \approx 2.82 \times 10^{12}$, which adequately demonstrates the diversity of the SAAE search space compared with manually designed encoders.

In practical applications, SAAE conducts a layer-by-layer search of N-layer modules using specified scenario data. It iteratively selects the optimal operation from multiple candidate operations within each functional module, thereby determining the search result of the current module. This process continues until the complete structure of the acoustic encoder is obtained. Subsequently, when combined with the downstream units of the corresponding end-to-end framework, an entire end-to-end ASR system is constructed to accomplish the final recognition task.

2.2 Differential search algorithm

Upon defining the search space illustrated in Figure 2, the challenge of selecting the optimal SAAE encoder structure from this extensive array of possibilities arises. Exhaustive enumeration or manual selection, based on empirical knowledge, proves prohibitively costly and impractical given the multitude of candidate encoder structures. To efficiently navigate this encoder structure, SAAE employs the differentiable architecture search algorithm known as Differentiable Architecture Search (DARTS). Originally proposed by Liu et al. for searching CNN model structures in image classification tasks. DARTS offers a more efficient alternative to other search algorithms, such as those based on reinforcement learning or evolutionary algorithms, which require continuous sampling and training. DARTS uses gradient backpropagation to independently learn the weights of candidate operations, thereby significantly improving search efficiency.

To enable differentiation operations within a discrete search space and facilitate gradient backpropagation, DARTS uses the softmax function to relax the entire search space into a continuous form as Equation 1.

$$x^{(i+1)} = \overline{o}^{(i,i+1)}\left(x^{(i)}\right) = \sum_{k=1}^{|O|} \frac{\exp\left(\alpha_k^{(i,i+1)}\right)}{\sum_{k'=1}^{|O|} \exp\left(\alpha_{k'}^{(i,i+1)}\right)} o_k^{(i,i+1)}\left(x^{(i)}\right), \quad (1)$$

where $|O|$ represents the total number of candidate operations between the two computing nodes $(i, i+1)$, $\overline{o}^{(i,i+1)}\left(x^{(i)}\right)$ denotes the computational result after the mixing of $x^{(i)}$, i.e., $x^{(i+1)}$. $o_k^{(i,i+1)}(\cdot)$ specifically refers to the k-th candidate operation between the two computing nodes $(i, i+1)$, and $\alpha_k^{(i,i+1)}$ represents the corresponding learnable structural parameter. After calculating the softmax on the structural parameter α, each candidate operation is assigned a respective weight, and a weighted sum is conducted on multiple candidate operations, thereby making several distinct candidate operations continuous, allowing further differentiation of the search space and enabling gradient backpropagation. During the search process, SAAE utilizes a two-level optimization strategy, alternating between optimizing the structural parameters α and the model's parameters W based on the target scenario data, to ultimately obtain an adaptive optimal structural parameter α for the given scenario. Once the search is completed, SAAE utilizes the optimal structural parameter α to select the operation with the highest corresponding weight from the multiple candidate operations for each module in the search space, thereby obtaining the acoustic decoder structure. Subsequently, this encoder is combined with the corresponding downstream units for joint retraining, ultimately yielding a well-trained end-to-end ASR system.

Although DARTS effectively improves search efficiency, it introduces a certain degree of search bias. During the search process, SAAE uniformly optimizes all candidate operation parameters in the entire search space; however, the search results used for retraining after the search is completed represent only a subset of the entire search space. The inconsistency between the two can cause some search bias, leading to the actual encoder structure obtained from the search not necessarily being the optimal one in the current search space. To address this challenge, the Gumbel reparameterization trick [34] is used to modify the continuous relaxation method in Equation 1 to minimize the discrepancy between the search and retraining. First, let $G_k^{(i,i+1)} = -\lg\left(-\lg\left(U_k^{(i,i+1)}\right)\right)$ denote a random variable following the Gumbel distribution, where $U_k^{(i,i+1)}$ is a random variable following the uniform distribution. Given the relaxation temperature λ, based on $G_k^{(i,i+1)}$, the following relaxation strategy based on Gumbel sampling in the search space is obtained (Gumbel-softmax), as Equation 2.

$$\overline{o}^{(i,i+1)}\left(x^{(i)}\right) = \sum_{k=1}^{|O|} \frac{\exp\left[\left(\alpha_k^{(i,i+1)} + G_k^{(i,i+1)}\right)\backslash\backslash/\lambda\right]}{\sum_{k'=1}^{|O|} \exp\left[\left(\alpha_{k'}^{(i,i+1)} + G_{k'}^{(i,i+1)}\right)/\lambda\right]} o_k^{(i,i+1)}\left(x^{(i)}\right), \quad (2)$$

In comparison to the softmax-based relaxation strategy, the Gumbel-softmax introduces the concept of relaxation temperature. During the actual search process, by gradually reducing the relaxation temperature, the calculated weight distribution approaches "one-hot encoding." This ensures that the optimized parameter structure during the search process is closely aligned with that during retraining, thereby significantly reducing search bias. Moreover, sampling based on the Gumbel distribution still adheres to the weight distribution obtained using softmax, thereby preserving the optimization efficacy for the structural parameter. Section 4.3 compares the two search strategies and validates the effectiveness of the Gumbel sampling technique.

3 End-to-end ASR pre-training search strategy

In the training process, besides the structural and model parameters in the SAAE encoder, the three end-to-end frameworks include a multitude of parameters within their downstream units that necessitate training. For example, the CTC framework is the CTC classifier, the AED framework comprises the autoregressive decoder, and the RNN-T framework includes the prediction and joint networks. This adds complexity to the SAAE encoder structural search, as the outputs of the SAAE encoder may be severely underfitted when interfacing with the downstream units during the initial stages of model training. Consequently, providing accurate guidance to the SAAE structural search becomes challenging, potentially leading to suboptimal optimization in the early stages and affecting final performance.

To address this challenge, this paper proposes a pre-training initialization strategy for downstream units in end-to-end frameworks. Rather than initializing the downstream units randomly, a certain degree of pre-training is first applied to

these units. Subsequently, these pre-trained downstream units serve as guides for the SAAE encoder structure search. This enables the SAAE to receive more accurate guidance from the downstream units, thereby reducing performance loss attributed to their underfitting and yielding better search results. In addition, the probability of overfitting can be effectively reduced through pre-training.

The process of pre-trained SAAE structure search and retraining can be summarized as follows: (1) First, a pre-training model is constructed based on the selected framework, where encoder adopts a manually designed structure such as Transformer or Conformer based on the MHSA mechanism, while the downstream units are determined by the end-to-end model framework; (2) The pre-training model is initialized to a certain extent using the target scenario dataset, after which the manually designed encoder is removed, retaining only the model parameters corresponding to the downstream units; (3) The pre-trained downstream units serve as the initialization for the downstream units of the SAAE encoder, and the structure of each module in the SAAE encoder is then searched layer by layer; (4) The SAAE encoder structure derived from the search results is combined with the downstream units to build the complete end-to-end ASR model, which is then retrained until convergence is achieved; (5) The retrained end-to-end ASR model is integrated with modules such as voice activity detection and feature signal processing to construct a complete end-to-end ASR system, which is subsequently tested and evaluated on the corresponding dataset to obtain recognition performance metrics such as error rates.

4 Experimental analysis

4.1 Experimental datasets and evaluation metrics

To validate the effectiveness of SAAE, experiments were conducted using three commonly used Chinese and English datasets: Aishell-1 [35], HKUST [36], and SWBD [37].

Aishell-1 is a comprehensive Mandarin speech dataset comprising 178 h of speech data. It consists of three subsets: train, dev, and test. Each speech segment contains utterances from a single speaker. The dataset covers 11 different task scenarios, including smart home, autonomous driving, and industrial production. Recordings were made using three different devices (high-fidelity microphone, Android phone, and iOS phone) in a quiet indoor environment.

HKUST is a Chinese Mandarin telephone speech dataset recorded under the supervision of the Hong Kong University of Science and Technology. It includes 200 h of speech data divided into training and test subsets. Each telephone speech segment contains utterances from two speakers engaging in daily conversations spanning various topics such as society, economics, entertainment, and sports.

SWBD: is a series of English telephone speech datasets released by the Linguistic Data Consortium (LDC). For this study, the 300-hour Switchboard-I (LDC97S62) dataset was selected for training. This dataset includes more than 70 diverse topics of daily conversations, with each conversation involving two speakers. Additionally, the Hub5′00 dataset, a subset of the SWBD series, was used as the test set, comprising approximately 11 h of data from the swbd1 and callhm subsets.

Overall, these three datasets exhibit significant differences in content, topics, languages, and speakers, providing comprehensive validation of SAAE's performance in adaptively designing acoustic encoders based on given scenario-specific speech data across diverse contexts.

For the two Chinese datasets (Aishell-1 and HKUST), the Character Error Rate (CER) was selected as the evaluation metric, which was calculated as the ratio of incorrectly recognized characters to the total number of characters. For the SWBD English dataset, the Word Error Rate (WER) was chosen as the evaluation metric. Lower CER values indicate better system performance. WER is calculated as the ratio of incorrectly recognized words to the total number of words. Lower CER or WER values indicate better system performance, aligning with current mainstream practice in the academic community.

4.2 Baseline models and implementation methodology

This study used two common manually designed acoustic encoders: Transformer and Conformer, chosen as baselines for comparison with SAAE. The Transformer is the most widely used encoder structure based on the MHSA mechanism and has found extensive application. The Conformer, after incorporating CNN mechanisms, represents the best-performing encoder structure for end-to-end ASR tasks. The optimal performance of the manually designed baselines on the three datasets used in this study was achieved by the Conformer encoder. The structures of the two baseline encoders were manually adjusted to examine the influence of different manual encoder structures on system performance. In the Transformer encoder, the primary functional unit is the MHSA. Hence, this study provides various specifications of attention heads for MHSA, such as 4, 8, and 16, denoted as H4, H8, and H16, respectively. Apart from MHSA, the Conformer encoder emphasizes the modeling of context information using CNN in addition to MHSA. Therefore, this study provided different combinations of the number of MHSA attention heads and the size of CNN convolutional kernels, denoted as HxCy, representing the combination of attention heads and convolutional kernels.

In terms of the specific implementation of the models, the open-source end-to-end ASR training tool ESPnet [38], based on PyTorch, was utilized for model training. Hyperparameters related to model design and training according to the optimal settings provided by ESPnet developers. For both the baseline model and SAAE, the attention dimension was set to 256, and the number of encoder layers was set to 8. The hidden layer mapping dimension of the FFN module in the baseline model was fixed at 1,024. The CNN module structure used in Conformer and SAAE was consistent with the design by Gulati et al. In the CTC framework, the output dimension of the CTC classifier is aligned with the number of output labels in the dataset. The AED employed a four-layer Transformer decoder, whereas the prediction network of the RNN-T used a one-layer LSTM. Specific model parameters adhered to the optimal settings

TABLE 1 Comparison of CER of SAAE encoder and handcrafted encoder under CTC framework.

Model structure	Settings	Aishell-1			HKUST		Hub5′00		
		Parameter count(M)	dev (%)	Test (%)	Parameter count(M)	Test (%)	Parameter count(M)	swbd1 (%)	Callhm (%)
Transformer	H4	8.18	7.1	7.7	8.18	24.1	8.18	13.4	24.7
	H8	8.19	6.9	7.5	8.19	23.7	8.19	13.8	25.2
	H16	8.19	6.9	7.4	8.19	23.8	8.19	13.6	25.1
Conformer	H4C7	9.69	5.8	6.5	9.69	23.4	9.69	11.8*	22.1
	H4C15	9.75	6.0*	6.6*	9.75	22.8*	9.75	12.0	22.4
	H4C31	9.93	6.5	7.2	9.93	23.7	9.93	11.8*	22.0*
	H8C15	9.75	6.0*	6.7	9.75	22.8*	9.75	12.3	22.9
	H16C15	9.75	6.2	6.8	9.75	23.4	9.75	12.2	22.8
Random search	searched	9.95	6.3	6.0	9.15	23.9	9.65	12.4	23.2
SAAE (No pre-training)	searched	9.43	5.9	6.5	9.83	22.8	9.33	11.8	22.1
SAAE (softmax)	searched	9.23	6.1	6.8	9.93	23.3	9.57	12.1	22.3
SAAE	searched	9.12	5.6	6.1	9.07	22.1	9.78	11.5	21.5

provided by ESPnet. All experiments were conducted on a computer equipped with a single Nvidia RTX4090 GPU, a 32-core Intel (R) Core i9-13900K CPU, and 32 GB RAM.

In this study, SAAE used pre-training and the Gumbel-Softmax technique by default, with the temperature factor in Gumbel-Softmax exhibiting exponential decay. The effectiveness of SAAE using pre-training and Gumbel-Softmax was compared with that of SAAE without pre-training and using softmax. Furthermore, to independently evaluate the effectiveness of the search algorithm, a comparison with the random search algorithm was conducted. The specific approach involved uniformly sampling five model samples from the SAAE search space and selecting the best-performing model on the development set as the experimental result for random search. Each dataset and each end-to-end framework underwent an independent random search.

4.3 Experimental results and analysis

Tables 1–3 present the experimental results under the CTC, AED, and RNN-T frameworks, including the CER on the Chinese test dataset and the WER on the English test dataset. The best results for each dataset are highlighted in bold, with "*" indicating the best result obtained by the manually designed baseline on that dataset.

By comparing the results of the manually designed encoders with various structural parameters across the three frameworks, it can be observed that due to the different acoustic characteristics present in different speech scenarios, an acoustic encoder structure suitable for one scenario may not necessarily be suitable for another. A fixed encoder structure cannot achieve the lowest error rate for all datasets. Taking the AED framework as an example, among all manual baselines, the Conformer encoder with the H16C15 structure achieved the lowest error rate on the Aishell-1 dataset, but this structure did not deliver the optimal performance on the HKUST and SWBD datasets. This highlights the necessity of designing encoder structures based on the acoustic characteristics of target speech scenarios. Furthermore, it is evident that by manually experimenting with various encoder structures, relatively lower error rates can be achieved in the target scenarios, with the error rates at the "*" points in Tables 1–3 noticeably lower than those of a fixed manual encoder structure.

While manually designing various encoder structures improves recognition performance to some extent, SAAE offers a better method for adaptively designing acoustic encoders based on the target dataset. The advantages of SAAE are evident in two aspects: First, the SAAE search space provides a more diverse set of candidate encoders, ensuring that SAAE can explore a wider range of encoder structures, thereby achieving lower character error rates. As illustrated in Tables 1–3, under the three common end-to-end frameworks, the character error rates obtained by SAAE are significantly lower than those of the manually designed encoder baselines. Additionally, from the parameter count column, it can be observed that the encoder structures obtained by SAAE, while having either a smaller parameter count or a marginal

TABLE 2 Comparison of CER of SAAE encoder and handcrafted encoder under AED framework.

Model structure	Settings	Aishell-1			HKUST		Hub5′00		
		Parameter count (M)	dev (%)	Test (%)	Parameter count (M)	Test (%)	Parameter count (M)	swbd1 (%)	Callhm (%)
Transformer	H4	8.18	5.7	6.2	8.18	22.8	8.18	8.4	17.6
	H8	8.19	6.2	6.5	8.19	22.6	8.19	8.5	17.2
	H16	8.19	5.3	5.9	8.19	22.6	8.19	8.6	18.0
Conformer	H4C7	9.69	5.3	5.9	9.69	21.5	9.69	8.1	16.0
	H4C15	9.75	5.1*	5.7*	9.75	21.4*	9.75	8.2	16.3
	H4C31	9.93	5.2	5.8	9.93	21.8	9.93	8.0*	15.9*
	H8C15	9.75	5.2	5.7*	9.75	21.6	9.75	8.1	16.3
	H16C15	9.75	5.1*	5.7*	9.75	22.0	9.75	8.2	16.5
Random search	searched	9.37	5.4	6.2	9.52	22.2	9.09	8.3	17.1
SAAE (No pre-training)	searched	9.43	5.1	5.6	9.93	21.4	9.52	8.0	16.1
SAAE (softmax)	searched	9.73	5.2	5.7	9.23	21.6	9.43	8.2	16.5
SAAE	searched	9.05	4.8	5.3	9.11	21.0	9.82	7.7	15.3

increase, achieve better WER results. Furthermore, under otherwise unchanged conditions, the model's computational complexity is correlated with its parameter count. Therefore, the computational efficiency of SAAE models is comparable to that of other models, and empirical observations confirm this perspective. Thus, in practical applications, SAAE models do not consume excessive computational resources compared to other models. Across all datasets, compared with the best results of the manually designed encoders (*), SAAE achieved an average relative reduction of 5% in the character error rate.

Second, by employing a differentiable search algorithm, the design cost of SAAE encoders is much lower than that of the manual encoder design strategy. Table 4 compares the time costs between the search process of SAAE and the training of manually designed encoders. The training time for the manual baselines is the cumulative training time for the aforementioned eight different manually designed baseline encoders. The strategy of manually designing multiple encoders requires the expensive time cost of training and evaluating each encoder structure individually. However, SAAE effectively reduces the design cost of the encoder by using a differentiable search algorithm. The time expenses of SAAE consist of pre-training, search, and re-training, and even the combined time for these three components is considerably lower than the strategy of training multiple manual baselines. Compared with the strategy of manually designing various encoder structures, SAAE reduces the time cost by over 75%.

In summary, SAAE provides a scenario-adaptive acoustic encoder design method that is superior to manual methods in terms of both performance and efficiency.

Given that the search space used by random search aligns with that of SAAE, the performance improvement of SAAE compared with the results obtained by random search can be entirely attributed to the search algorithm used by SAAE. This indicates the effectiveness of the differentiable search algorithm employed to select encoder structures in this study. When comparing the experimental results without using end-to-end pre-training, SAAE demonstrated comprehensive performance improvements. Moreover, when compared with the AED and RNN-T frameworks, the CTC framework exhibits the smallest performance degradation when pre-training is not used. This might be attributed to CTC having the fewest downstream unit parameters among the three frameworks, necessitating the least number of parameters for pre-training, thus having the least impact on the CTC framework. This finding aligns with the analysis presented in Section 3.

In comparing the experimental results using the Gumbel-softmax and softmax relaxation strategies, it becomes evident that the direct use of softmax leads to a significant performance degradation compared with the use of Gumbel-softmax. This highlights the necessity of using Gumbel-softmax during the search process. Figure 3 depicts the variation of the loss function on the training and validation sets when using the CTC framework with softmax and Gumbel-softmax for the encoder structure search

TABLE 3 Comparison of CER of SAAE encoder and handcrafted encoder under RNN-T framework.

Model structure	Settings	Aishell-1			HKUST		Hub5′00		
		Parameter count (M)	dev (%)	Test (%)	Parameter count (M)	Test (%)	Parameter count (M)	swbd1 (%)	Callhm (%)
Transformer	H4	8.18	6.5	7.2	8.18	27.8	8.18	11.3	20.5
	H8	8.19	6.3	6.8	8.19	27.9	8.19	11.1	20.4
	H16	8.19	6.3	7.0	8.19	27.7	8.19	11.5	20.9
Conformer	H4C7	9.69	6.0*	6.8	9.69	27.3	9.69	10.3*	20.2
	H4C15	9.75	6.0*	6.8	9.75	26.6*	9.75	10.5	19.6
	H4C31	9.93	6.1	6.9	9.93	26.8	9.93	10.7	20.1
	H8C15	9.75	6.0*	6.7*	9.75	27.0	9.75	10.7	19.6*
	H16C15	9.75	6.0*	6.7*	9.75	27.2	9.75	10.5	20.0
Random search	searched	9.71	6.2	6.9	9.83	27.2	9.47	10.9	20.5
SAAE (No pre-training)	searched	9.18	6.0	6.4	9.81	26.5	9.32	10.6	19.8
SAAE (softmax)	searched	9.03	6.1	6.8	9.39	27.1	9.54	10.2	19.4
SAAE	searched	9.85	5.6	6.2	9.33	25.9	9.72	9.8	18.8

TABLE 4 Comparison of the training time cost of the SAAE encoder and the handcrafted encoder.

Method	Aishell-1			HKUST			SWBD		
	CTC	AED	RNN-T	CTC	AED	RNN-T	CTC	AED	RNN-T
Manual baseline training	156.5	195.6	231.6	210.9	265.1	316.1	476.6	598.1	725.7
SAAE pre-training	2.1	2.5	3.2	2.8	3.4	4.2	6.3	7.9	9.2
SAAE Search	12.5	15.0	18.7	16.4	20.5	24.5	37.1	46.6	60.4
SAAE re-training	20.6	25.2	31.4	27.3	34.1	41.1	62.0	77.6	94.0
Total time of SAAE	35.2	42.7	53.3	46.5	58.0	69.8	105.4	132.1	163.6

on the Aishell-1 dataset. Here, "train_loss" denotes the training set error, while "validation_loss" refers to the validation set error. When using softmax, after reaching a certain point in training, the validation set loss significantly deviates from the training set loss, indicating search bias due to the use of softmax. Gumbel-softmax effectively mitigates this phenomenon, as the performance of the validation set consistently improves in conjunction with the training set for the entire search process. This implies that Gumbel-softmax significantly reduces the deviation between the training and validation sets during the search process, which aligns with the analysis in Section 2.2.

In practical applications, aside from factors such as speech content and language, ASR systems often encounter speech data contaminated with noise and reverberation. These noisy speech data notably differ in acoustic characteristics from clean speech. In this study, reverberation and noise were added to the train and dev data of Aishell-1, resulting in a new noisy dataset termed Aishell-Noisy. Initially, reverberation at three distances (1, 2, and 5 m) was randomly added to the original clean speech. Followed by the superposition of this reverberated speech with the open-source WHAM noise dataset [39]. The signal-to-noise ratio was uniformly and randomly distributed between 0 and 20 dB. The resulting

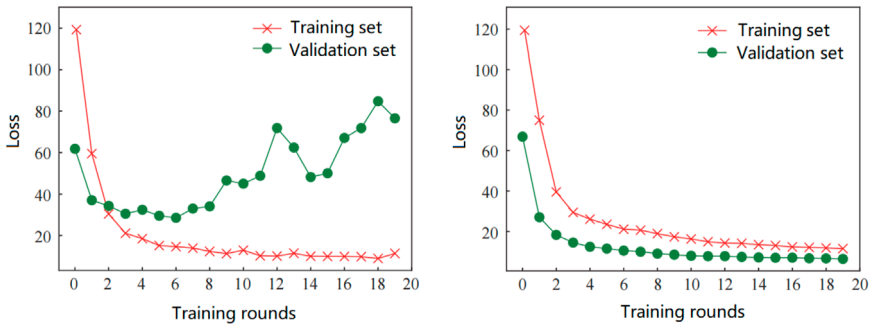

FIGURE 3
Loss function variation curve on Aishell-1: **(A)** use softmax, **(B)** use Gumbel-softmax.

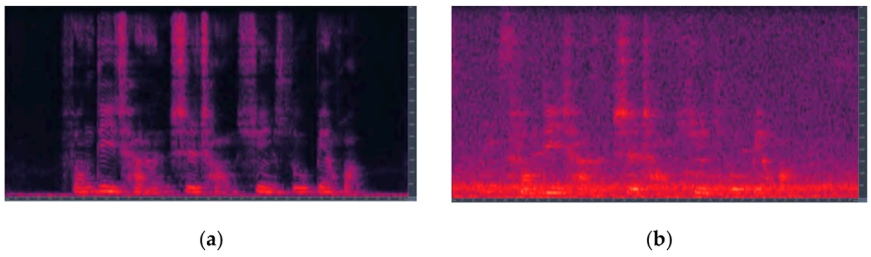

FIGURE 4
Examples of changes in speech spectral features before and after noise were added to Aishell-1 speech: **(A)** clean speech, **(B)** speech with 5 m reverberation, and 20 dB noise added.

Aishell-Noisy dataset maintains the same speaker information, speech content, and language information as the original Aishell-1 dataset, with the only distinction being the presence of noise.

Figure 4 illustrates the spectrogram of the speech sample BAC009S0002W0150 in the dataset, both before and after the addition of 5 m reverberation and noise with a signal-to-noise ratio of 20 dB. Despite identical speech content, the spectrograms before and after noise addition present distinct spectral characteristics. Consequently, even when the speech content is identical, noisy speech and clean speech should be viewed as belonging to two different speech scenarios, and the most suitable acoustic encoder structure may differ entirely in these two scenarios.

To verify the aforementioned hypothesis, further experiments were conducted on the Aishell-Noisy dataset using the AED model framework. Table 5 presents the recognition performance of various encoder structures trained on Aishell-Noisy. When faced with noisy scenarios, compared to manually crafted baselines, the encoder structures obtained through search on clean speech did not yield significant performance improvements and failed to achieve better recognition performance on noisy speech than the optimal manual baseline (H4C31). Conversely, conducting an encoder structure search directly on noisy speech proved to be highly effective in enhancing recognition performance on the noisy test set. In comparison to the encoder structures obtained through search on clean speech and the optimal manual baseline, the CER was improved by over 10%. This suggests that by treating noisy speech as a distinct acoustic scenario, SAAE offers a novel approach to achieving robust speech recognition. SAAE can be utilized to design acoustic encoders with enhanced robustness and noise resistance,

TABLE 5 Comparison of the CER of different SAAE encoders with handcrafted encoders on Aishell-Noisy.

Encoder	Settings	Aishell-noisy dev (%)
Transformer	H4	20.2
Transformer	H8	19.8
Transformer	H16	20.1
Conformer	H4C7	19.1
Conformer	H4C15	17.3
Conformer	H4C31	17.1*
Conformer	H8C15	18.0
Conformer	H16C16	17.5
SAAE	Aishell-1 search obtained	17.6
SAAE	Aishell-Noisy search obtained	15.0

TABLE 6 Comparison of CER on Aishell-1 between ablation and full search space.

Model structure	CTC		AED		RNN-T	
	dev (%)	test (%)	dev (%)	test (%)	dev (%)	test (%)
Transformer (best)	6.8	7.4	5.5	5.8	6.3	6.7
Conformer (best)	5.8	6.5	5.1	5.7	6.1	6.6
Fixed MHSA4	5.9	6.6	5.0	5.5	5.9	6.4
Fixed CNN15	5.8	6.3	5.2	5.6	6.1	6.8
Fixed FFN1024	5.6	6.1	4.8	5.3	5.6	6.2
Stack the same modules	5.7	6.3	5.1	5.7	5.9	6.6
SAAE	5.6	6.1	4.8	5.3	5.6	6.2

thereby improving the recognition performance of ASR systems in noisy environments.

4.4 Search space ablation experiment

To further investigate the impact of the SAAE search space on performance, a series of ablation experiments were conducted on the Aishell-1 dataset focusing on the SAAE search space. Two specific approaches were employed: (1) fixing three modules as particular candidate operations and conducting searches only on the remaining two modules. For instance, SAAE-MHSA4 refers to fixing MHSA as the MHSA4 operation and searching for CNN and FFN; (2) searching for a single encoder module, and subsequently constructing the entire encoder by stacking N layers of the same module. Table 6 presents the results of the ablation experiments and the experiments with manual baselines, where Transformer and Conformer achieved the best performance under the corresponding frameworks, i.e., the results marked with "*" in Tables 1–3.

From Table 6, it is evident that restricting the search space by fixing the MHSA and CNN modules to specific subsequent operations or by directly stacking multiple layers of the same module leads to an increase in the CER. This result underscores the importance of SAAE in providing multiple candidate operations for each module in the search space. Furthermore, it emphasizes the importance of independently designing the structure of multiple layers. A noteworthy observation is that when the FFN layer is fixed to FFN1024, the performance remains stable compared to the complete search space. This observation aligns with the conclusions drawn from the manual design of Transformer and Conformer, indicating that for the FFN layer, a higher activation dimension results in stronger modeling capability and better performance.

5 Conclusion

To systematically design high-performance acoustic encoders tailored to the acoustic characteristics of specific scenarios, we proposed a scene-adaptive acoustic encoder method, the SAAE, which leverages neural network architecture search techniques. Through differentiable optimization methods, SAAE comprehensively analyzes speech data in specific scenarios to uncover acoustic characteristics across multiple dimensions such as emotion, language, noise, and channel variations. Subsequently, based on these identified acoustic features, SAAE designs a high-performance acoustic encoder adapted to the target scenario. Experimental results demonstrate that SAAE effectively enhances recognition performance across diverse acoustic scenarios, yielding lower error rates than existing methods. Although Xiaomi's recently released Zipformer [40] model shows better recognition rates on the Aishell-1 dataset, with a parameter count of 157.3 M, the SAAE model presented here has a parameter count of only 9.05 M. This smaller parameter size, while maintaining good recognition rates, is better suited for resource-constrained devices. Therefore, the SAAE approach is an effective method for designing high-performance acoustic encoders tailored to specific acoustic scenarios.

However, the algorithm presented in this paper has only been experimentally validated on Mandarin and English, and its results may not necessarily apply to other languages due to differing acoustic characteristics. To enhance the algorithm's broader applicability, it is evidently necessary to conduct experimental and research analyses on a wider range of languages. This will be one of our primary research directions moving forward.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

YW: Conceptualization, Formal Analysis, Investigation, Validation, Writing–original draft. XL: Formal Analysis, Validation, Writing–review and editing. FG: Formal Analysis, Validation,

Writing–review and editing. TL: Formal Analysis, Resources, Writing–review and editing. CL: Conceptualization, Investigation, Writing–review and editing.

Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. This research was funded in part by the "Special Support Program for High-level Talents of China Southern Power Grid" (Grant No. 202301011056). The funder was not involved in the study design, collection, analysis, interpretation of data, the writing of this article, or the decision to submit it for publication.

Conflict of interest

Authors YW, XL, FG, TL, and CL were employed by Guangdong Power Grid Co., Ltd., Jiangmen Power Supply Bureau.

Publisher's note

References

1. Malik M, Malik MK, Mehmood K, Makhdoom I. Automatic speech recognition: a survey. *Multimedia Tools Appl* (2021) 80:9411–57. doi:10.1007/s11042-020-10073-7

2. Shan C, Zhang J, Wang Y, Xie L. Attention-based end-to-end speech recognition on voice search. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE (2018) p. 4764–8.

3. Tulshan AS, Dhage SN. Survey on virtual assistant: google assistant, siri, cortana, alexa. In: Advances in signal processing and intelligent recognition systems: 4th international symposium SIRS 2018. Singapore: Springer (2019) p. 190–201.

4. de Vos H, Verberne S. Challenges of applying automatic speech recognition for transcribing eu parliament committee meetings: a pilot study. In: *Proceedings of the second ParlaCLARIN workshop* (2020) p. 40–3.

5. Jiang J, Wang HH. Application intelligent search and recommendation system based on speech recognition technology. *Int J Speech Technology* (2021) 24:23–30. doi:10.1007/s10772-020-09703-0

6. Bingol MC, Aydogmus O. Performing predefined tasks using the human–robot interaction on speech recognition for an industrial robot. *Eng Appl Artif Intelligence* (2020) 95:103903. doi:10.1016/j.engappai.2020.103903

7. Variani E, Bagby T, McDermott E, Bacchiani M (2017) End-to-end training of acoustic models for large vocabulary continuous speech recognition with tensorflow. 1641, 5. doi:10.21437/interspeech.2017-1284

8. Wassink AB, Gansen C, Bartholomew I. Uneven success: automatic speech recognition and ethnicity-related dialects. *Speech Commun* (2022) 140:50–70. doi:10.1016/j.specom.2022.03.009

9. Cave R, Bloch S. The use of speech recognition technology by people living with amyotrophic lateral sclerosis: a scoping review. *Disabil Rehabil Assistive Technology* (2023) 18(7):1043–55. doi:10.1080/17483107.2021.1974961

10. Murali PK, Kaboli M, Dahiya R. Intelligent in-vehicle interaction technologies. *Adv Intell Syst* (2022) 4(2):2100122. doi:10.1002/aisy.202100122

11. Ren Y, Liu J, Tan X, Zhang C, Qin T, Zhao Z, et al. SimulSpeech: end-to-end simultaneous speech to text translation. In: *Proceedings of the 58th annual meeting of the association for computational linguistics* (2020). p. 3787–96.

12. Chi OH, Denton G, Gursoy D. Artificially intelligent device use in service delivery: a systematic review, synthesis, and research agenda. *J Hospitality Marketing and Management* (2020) 29(7):757–86. doi:10.1080/19368623.2020.1721394

13. Xia X, Ma Y, Luo Y, Lu J. An online intelligent electronic medical record system via speech recognition. *Int J Distributed Sensor Networks* (2022) 18(11):155013292211344. doi:10.1177/15501329221134479

14. Jiang Y, Li X. Intelligent online education system based on speech recognition with specialized analysis on quality of service. *Int J Speech Technology* (2020) 23:489–97. doi:10.1007/s10772-022-10004-x

15. Liu J, Yu H, Huang H. End-to-End deep convolutional neural network speech recognition. *Comput Appl Softw* (2020) 37(4):192–6.

16. Zilong W, Junfeng L, Zhang S, Wang H, Wang S. End-to-End speech recognition based on recurrent neural network. *Comput Digital Eng* (2019) 47(12):3099–106.

17. Tang H, Xue J, Han J. A multiscale forward attention model for speech recognition. *J Electronics* (2020) 48(7):1255–60. doi:10.3969/j.issn.0372-2112.2020.07.002

18. Guo J, Han J. An end-to-end speech recognition model combining RNN-T and BERT. *J Intell Comput Appl* (2021) 11(2):169–73.

19. Zhang K, Xiaofen Z. Robust speech recognition based on adaptive deep neural networks in complex environments. *J Comput Eng Sci* (2022) 44(6): 1105–13.

20. Vielzeuf V, Antipov G. Are E2E ASR models ready for an industrial usage? arXiv preprint: 2112.12572, 2021.

21. Yang W, Yan H. End-to-End Mandarin speech recognition with accents using hybrid CTC/attention architecture. *J Comput Appl Res* (2021) 38(3):755–9. doi:10.19734/j.issn.1001-3695.2020.02.0036

22. Jain M, Schubert K, Mahadeokar J, Yeh CF, Kalgaonkar K, Sriram A, et al. RNN-T for latency controlled ASR with improved beam search. *arXiv preprint:1911.01629* (2019). doi:10.48550/arXiv.1911.01629

23. Graves A, Fernández S, Gomez F, Schmidhuber J. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: *23rd international conference on Machine learning* (2006) p. 369–76.

24. Liu X, Song W, Chen X, Huan J, Li Z. BLSTMCTC speech recognition based on multikernel convolution fusion network. *J Comput Appl Softw* (2021) 38(11): 167–73.

25. Shen Y, Sun J. Lightweight Chinese speech recognition combined with transformer. *J Comput Appl Res* (2023) 40(2):424–9.

26. Liu H, Simonyan K, Yang Y. DARTS: differentiable architecture search. In: International conference on learning representations (2018).

27. So D, Le Q, Liang C. The evolved transformer. In: International conference on machine learning. Long Beach, CA: PMLR (2019) p. 5877–86.

28. Xiao Y, Yewei S, Guanying H, Ning X. Lightweight network construction based on neural network structure search. *Pattern Recognition Artif Intelligence* (2021) 34(11):1038–48.

29. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Aidan N, et al. Attention is all you need. In: 31st international conference on neural information processing systems (2017). p. 6000–10.

30. Yeliang L, Zhang E, Tang Z. Research on speech recognition based on hybrid attention mechanism. *J Comput Appl Res* (2020) 37(1):131–4. doi:10.19734/j.issn.1001-3695.2018.06.0492

31. Gulati A, Qin J, Chiu CC, Parmar N, Zhang Y, Yu J, et al. (2020). Conformer: convolution-augmented transformer for speech recognition. *arXiv preprint: 2005* 08100.

32. Sanger TD. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural networks* (1989) 2(6):459–73. doi:10.1016/0893-6080(89)90044-0

33. Zhu X, Zhang F, Gao L, Ren X, Hao B. Research on speech recognition based on residual networks and gated convolutional networks. *J Comput Eng Appl* (2022) 58(7):185–91. doi:10.3778/j.issn.1002-8331.2108-0265

34. Jang E, Gu S, Poole B. Categorical reparameterization with gumbel-softmax. *arXiv preprint: 1611.01144* (2016). doi:10.48550/arXiv.1611.01144

35. Bu H, Du J, Na X, Wu B, Zheng H. Aishell-1: an open-source Mandarin speech corpus and a speech recognition baseline. In: 20th conference of the oriental chapter of

the international coordinating committee on speech databases and speech I/O systems and assessment. Seoul, Korea, IEEE (2017).

36. Liu Y, Fung P, Yang Y, Cieri C, Huang S, Graff D. HKUST/MTS: a very large scale Mandarin telephone speech corpus. In: International symposium on Chinese spoken language processing. Berlin, Heidelberg: Springer (2006) p. 724–35.

37. Godfrey JJ, Holliman EC, McDaniel J. SWITCHBOARD: telephone speech corpus for research and development. In: IEEE international conference on acoustics, speech, and signal processing. San Francisco, CA: IEEE Computer Society (1992) p. 517–20.

38. Watanabe S, Hori T, Karita S, Hayashi T, Nishitoba J, Unno Y, et al. Espnet: end-to-end speech processing toolkit. *arXiv preprint: 1804.00015* (2018). doi:10.48550/arXiv.1804.00015

39. Maciejewski M, Wichern G, McQuinn E, Le Roux J. WHAMR!: noisy and reverberant single-channel speech separation. In: IEEE international conference on acoustics, speech and signal processing. Barcelona (2020) p. 696–700.

40. Yao Z, Guo L, Yang X, Kang W, Kuang F, Yang Y, et al. (2023) Zipformer: a faster and better encoder for automatic speech recognition. In: The twelfth international conference on learning representations.