



Conformer Space Neural Architecture Search for Multi-Task Audio Separation

Shun Lu[†], Yang Wang[†], Peng Yao[†], Chenxing Li, Jianchao Tan, Feng Deng,
Xiaorui Wang, Chengru Song

Kuaishou Technology Co., Beijing, China

{lushun, wangyang, yaopeng, lichenxing, jianchaotan}@kuaishou.com

Abstract

Multi-task audio source separation aims to separate the audios collected from the complex environment into three fixed types of signal sources. Existing methods like EAD-Conformer usually take a manually designed model to process the separation. These networks may be sub-optimal since it is hard for humans to train and test all possible architectures. Especially, it is natural to adopt different optimal sub-structures for decoding different types of signals, which, however, is very hard for humans to enumerate. In this paper, we quantitatively analyze the redundancy of the EAD-Conformer network and customize an effective and efficient search space. We propose an efficient K-path search method to search for the optimal architectures from the Conformer-based search space. We conduct a comprehensive search in terms of block numbers, head numbers, and channel numbers. Extensive experiments demonstrate that our searched architectures outperform existing methods in terms of efficiency and effectiveness.

Index Terms: multi-task audio source separation, neural architecture search

1. Introduction

As the preliminary step for audio signal processing, audio separation plays an important role in speech recognition, music information retrieval, and keyword spotting. The audio signals recorded in live broadcasts and short videos usually contain three major components: speech, music, and other background noises. To obtain the separated signals for downstream tasks, multi-task audio source separation (MTASS) [1] aims to separate three fixed types of sound sources from the mixture audios.

Significant progress has been made in MTASS. Complex-MTASSNet [1] adopts stacked convolutions to estimate the spectrum and a residual estimation network to refine the complex spectra of each track. MRX [2] uses a Bi-directional Long Short-Term Memory network (Bi-LSTM) to capture long-term dependencies and performs separation on different scales. EAD-Conformer [3] applies the Conformer block to extracting the local information and modeling the independence of different signals from a global view. To capture the different characteristics of speech, music, and noises, EAD-Conformer [3] shares the same encoder but uses three independent decoders.

However, as pointed out by recent works [4, 5], though the global information extracted by the self-attention module is of vital importance, it consumes the most computation and contains large redundancy, greatly hindering the network efficiency. For example, as shown in Figure 1, we first compute the number of parameters of the standard EAD-Conformer [3], which has 27M parameters. The model parameters and performance change little when removing the convolution module of each

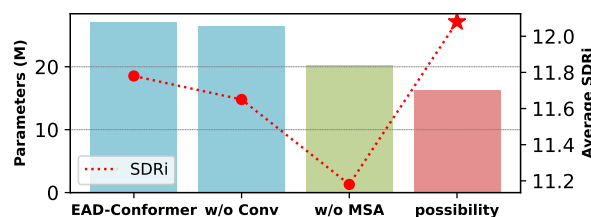


Figure 1: Model parameters and performance comparison. The first column represents the original EAD-Conformer [3], while 'w/o Conv' and 'w/o MSA' denotes the ablation EAD-Conformer after removing the convolution module and the multi-head self-attention module in each layer, respectively. 'possibility' means that we try to explore more lightweight architectures with higher performance (SDRi) in this paper.

conformer block, but a large decrease emerges once we omit the self-attention module of each conformer block, which reminds us that it is possible to obtain a lightweight architecture by slimming the self-attention module.

Moreover, hand-designed architectures have certain limitations and easily fall into the trap of the sub-optimum. Especially for the multi-task audio separation, it usually contains an audio feature encoder and several feature decoders for different separated audio tracks, whose structures may be identical if designed by humans. Among previous literature, the same structure used by each decoder may be inappropriate because re-constructing the audio of speech, music, and noise naturally requires distinct efforts. Nevertheless, it is challenging to design such customized architectures for different decoders depending on expertise, and thereby an effective search for elastic architectures on a well-defined search space is desired and necessary.

Fortunately, there are various Neural Architecture Search (NAS) methods proposed to automatically search high performance neural architectures with the manually designed search space and the manually chosen search algorithm. Early NAS methods adopt Reinforcement Learning (RL) [6, 7] or Evolution Algorithm (EA) [8] as the search engine and need to train many architectures from scratch to perform the architecture search, thus requiring huge computational costs even to thousands of GPU days. Conveniently, recent one-shot NAS methods [9] and differentiable NAS methods [10, 11] have reduced the search cost to several GPU days using the weight sharing mechanism [12], making it possible to readily apply NAS methods in many application fields. Some previous works have proved the effectiveness of NAS methods for the acoustic, such as in the acoustic scene classification [13–15], speech recognition [16], voice activity detection [17], and speaker verification [18], whereas there is no such try for multi-task audio separation.

In this paper, we propose a K-path search method to

[†] Equal contribution

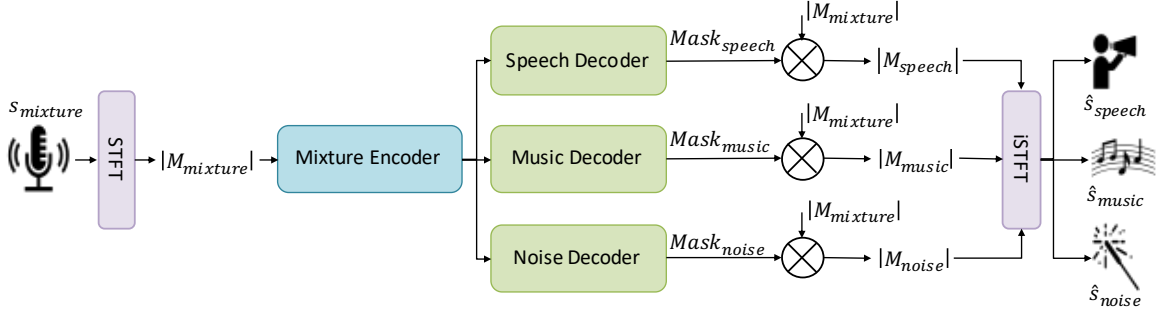


Figure 2: The overall process of MTASS. $|M_i|$ represents the STFT magnitude of signal. $Mask_i$ is a weight vector and is used to separate the mixture magnitude.

efficiently search for superb architectures in a pre-defined Conformer-based search space. Specifically, considering the redundancy of the self-attention module in the Conformer structure, we search for the optimal hidden dimension and the optimal number of attention heads in each block. To create a customized architecture for the encoder and each decoder, we also search for the appropriate depth for each part, i.e., the number of Conformer blocks. *Our framework is general enough and can be transferred to other tasks.* We summarize the main contributions as:

- By quantitatively analyzing the redundancy of the EAD-Conformer network, we customize an effective and efficient search space based on the Conformer block, inspiring the relevant model designs in the community.
- We further propose an efficient K-path search method to search for the optimal structure of the self-attention module in the Conformer block. Moreover, we also achieve an elastic depth for the encoder and each decoder.
- Comprehensive experiments are conducted to demonstrate that our searched architectures outperform state-of-the-arts with higher signal-to-distortion ratio [19] improvement (SDRi) but with fewer parameters.

2. System overview

We first introduce the processing procedure of the MTASS in Sec.2.1 and the loss function in Sec.2.2. Then we describe our constructed Conformer-based search space in Sec.2.3. Finally, we elaborate on our architecture search pipeline in Sec.2.4.

2.1. MTASS

Frequency domain-based audio separation can be formulated as follows: Give a mixture signal $M_{\text{mixture}}(t, f)$, we aim to recover N sources $S_i(t, f)$, $i = 1, \dots, N$, from the mixture.

$$\sum_{i=1}^N S_i(t, f) \rightarrow M_{\text{mixture}}(t, f), \quad (1)$$

where N is the number of source signals and $N = 3$ in this work. $M_{\text{mixture}}(t, f)$ and $S_i(t, f)$ represent the short-time Fourier transform (STFT) feature of the mixture and sources signal, respectively. In this paper, the signal is subjected to a STFT with a frame length of 1024 and a shift of 256. Its analysis window is a Hanning window.

In this experiment, the mixed STFT magnitude is acted as input, and the phase of the mixture is only used when recovering the waveform of the sources. Audio separation is converted

to the problem of recovering each source $|M_i(t, f)|$, given the magnitude of the mixed $|M_{\text{mixture}}(t, f)|$. As the spectrum to spectrum mapping is difficult [20], we turn to estimate a set of masks $Mask_i(t, f)$ [3] rather than estimate $|M_i(t, f)|$ directly,

$$\begin{aligned} Mask_i(t, f) &= g(|M_{\text{mixture}}(t, f)|, \theta), \\ |M_i(t, f)| &= Mask_i(t, f) \odot |M_{\text{mixture}}(t, f)|, \\ \hat{s}_i(t) &= \text{iSTFT}(|M_i(t, f)|, \angle M_{\text{mixture}}(t, f)), \end{aligned} \quad (2)$$

where $g(*)$ and θ represent the separation network and its parameters. \odot denotes the element-wise product. We use the inverse STFT (iSTFT) to recover the waveform $\hat{s}_i(t)$ with the phase of the mixture signal $\angle M_{\text{mixture}}(t, f)$. Accordingly, in the separation network, we utilize a Conformer encoder to refine the mixture feature and three Conformer decoders to get the separated audio masks. The overall process of the separation system is summarized in Figure 2.

2.2. Loss function

Following the EAD-Conformer [3], we adopt the L_1 loss to decrease the intra-class distance between the estimated signal $|M_i|$ and its corresponding clean signal $|S_i|$ and the discriminate loss to increase the inter-class distance between the estimated signal $|M_i|$ and other clean signals $|S_j|$ on the STFT magnitude,

$$\mathcal{L}_{\text{spec}} = \sum_{i=1}^N \mathcal{L}_1(|S_i| - |M_i|) - \lambda \sum_{i=1}^N \mathcal{L}_1(|S_i| - \sum_{j \neq i}^N |M_j|), \quad (3)$$

where λ is a hyper-parameter and we use $\lambda = 0.1$ in practice.

To further improve the generalization ability of the network, we also introduce the source-to-noise ratio (SNR) loss on the time domain,

$$\mathcal{L}_{\text{time}} = \sum_{i=1}^N \text{SNR}(\hat{s}_i, s_i), \quad (4)$$

where \hat{s}_i (Eq.2) and s_i are the estimated and clean waveform of separated signals.

2.3. Customized Conformer-based search space

Vanilla Conformer [21] is constructed by stacking M identical Conformer blocks, each of which includes the Multi-head Self-Attention (MSA) module and the Convolution (Conv) module sandwiching between two Feed Forward Network (FFN) modules. The Conv module can gather local neighborhood information effectively, and the MSA module is designed to capture

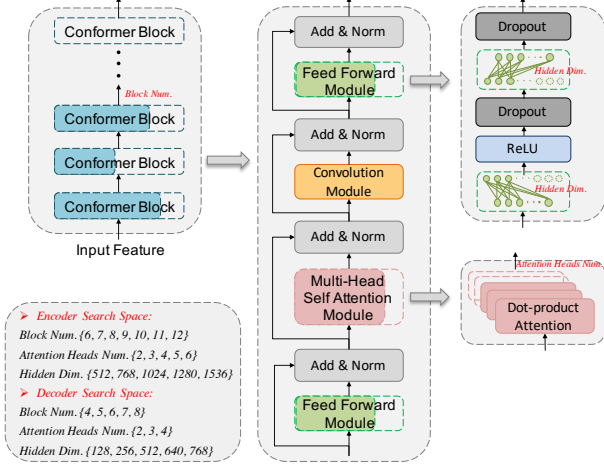


Figure 3: Our proposed conformer-based search space. We jointly search in both macro and micro levels of conformer structure for the encoder and three decoders.

global long-range dependencies. Given an input x_i to the i -th Conformer block, the output x_{i+1} can be formulated as below:

$$\begin{aligned}
 \tilde{x}_i &= x_i + \frac{1}{2} \text{FFN}(x_i), \\
 x'_i &= \tilde{x}_i + \text{MSA}(\tilde{x}_i), \\
 x''_i &= x'_i + \text{Conv}(x'_i), \\
 x_{i+1} &= \text{LayerNorm}\left(x''_i + \frac{1}{2} \text{FFN}(x''_i)\right).
 \end{aligned} \tag{5}$$

In this work, we jointly search for the hidden dimension of FFN, the attention heads number, and the elastic depth for the encoder and three decoders to discover a lightweight and effective architecture. Our search space is summarized in Figure 3.

Elastic hidden dimension for FFN. FFN module consists of two fully connected layers with an activation function in the middle, and we use the ReLU function in this work. We focus on searching for the optimal hidden dimension for the FFN module for both encoder and three decoders.

Elastic attention heads for MSA. MSA contains parallel scaled dot-product attention heads, and their outputs are concatenated to the subsequent layers. Previous studies have pointed out that MSA produces more powerful features than single-head self-attention. Therefore, our goal is to customize a suitable number of attention heads for each layer of the encoder and three decoders.

Elastic depth for the encoder and three decoders. The network needs appropriate depth to provide sufficient representation ability, and it requires distinct efforts to re-construct different source signals. Thus we search for the appropriate depth for the encoder and three decoders.

2.4. Search pipeline

Our search pipeline mainly includes two sequential phases: *K-path supernet training* and *evolutionary search*. Traditional one-shot NAS methods usually train a supernet \mathcal{A} with weights \mathcal{W} and then efficiently evaluate sub-model structures α with a heuristic algorithm by inheriting the pre-trained weights \mathcal{W}^*

from the supernet. The goal of NAS is to find the optimal sub-model α^* with top performance \mathcal{P} from this supernet:

$$\alpha^* = \arg \max_{\alpha} \mathcal{P}(\mathcal{A}(\mathcal{W}^*, \alpha)). \tag{6}$$

K-path supernet training. Motivated by ViT-ResNAS [22], we adopt a K-path sampling method to efficiently train the supernet. Specifically, for each batch data, we randomly split the batch data into k parts and generate k masks for each Conformer block. By assembling the different masks in each Conformer block, we can obtain k paths in the supernet. We forward and backward the supernet with these masks, amounting to an efficient training of random-generated k paths in parallel with the training loss $\mathcal{L}_{\text{train}}$, which can be formulated as below:

$$\mathcal{W}^* = \arg \min_{\mathcal{W}} \mathcal{L}_{\text{train}}(\mathcal{A}(\mathcal{W}, [\alpha_1, \dots, \alpha_k])). \tag{7}$$

Such K-path sampling strategy covers more sub-models with different configurations than single-path one-shot methods [23, 24] and thus is more efficient and effective.

Evolutionary search. After pre-training the supernet, we adopt the evolutionary algorithm NSGA-II [25] to generate candidate sub-models, which are required to meet the pre-defined resource constraints (e.g., the number of model parameters in our experiments). We aim to discover the optimal sub-model α^* , whose resource is no more than resource target $\mathcal{C}_{\text{target}}$ and then Eq.6 can be re-formulated as below:

$$\alpha^* = \arg \max_{\alpha} \mathcal{P}(\mathcal{A}(\mathcal{W}^*, \alpha)), \quad \text{s.t. } \mathcal{C}_{\alpha} \leq \mathcal{C}_{\text{target}}. \tag{8}$$

3. Experiments

3.1. Experimental setup

We search for the best architecture on the MTASS dataset [1], which contains 20,000 training data, 1000 validation data, and 1000 test data. This dataset includes three types of audio: speech, music, and noise. All of them are collected from aishell1 [33], DiDiSpeech [34], DSD100 [35], and DNS challenge [36]. The input mixture is generated by summing up the three types of audios with varying SDR, and all audios clips are down-sampled to 16kHz. We randomly extract continuous audios with the length of 6 seconds from the whole audios for the training and use 10-seconds (almost the entire audio length) continuous audios to calculate the final results for evaluation like previous works [3].

During the supernet training, we set k as the number of GPU cards. Other training configurations are kept the same for both the supernet and our searched architectures. Specifically, we train our network by 200 epochs, of which L_{spec} is utilized in the first 135 epochs and L_{time} is used in the last 65 epochs. We employ the Adam optimizer with the maximum learning rate 1e-3 and the minimum learning rate 1e-4. The Cosine scheduler is applied following the 10 epochs of linear warm-up. We use the clip-norm method with a clip rate of 5.0. We set the weight decay as 1e-5 and the training batch size as 80. Both the supernet training and sub-models training spend 20 hours on a machine with 8 NVIDIA Tesla V100 GPU cards, and we implement our code using the Pytorch 1.8 framework.

3.2. Comparison with the state-of-the-arts

We show three searched optimal architectures with different size of parameters, namely ConformerNAS-A, ConformerNAS-B, and ConformerNAS-C. The comparisons with previous

Table 1: Comparison with previous state-of-the-arts on the speech, music, and noise signal tracks. RTF means the processing time consumption per second (real-time factor) on GPU.

Method	Parameters (M)	FLOPs (G)	RTF (GPU)	SDRi (dB)			Avg.
				Speech	Music	Noise	
GCRN-RI [26]	9.88	2.5	0.031	9.11	5.76	5.51	6.79
GCRN-cRM [26]	9.88	2.5	0.031	8.73	6.25	6.50	7.16
Demucs [27]	243.32	5.6	0.006	9.93	6.38	6.29	7.53
D3Net [28]	7.93	3.5	0.002	10.55	7.64	7.79	8.66
Conv-TasNet [29]	5.14	5.2	0.017	11.80	8.35	8.07	9.41
DCCRN [30]	3.70	14.5	0.018	11.24	9.15	8.80	9.73
Sepformer [31]	25.75	77.7	0.052	11.33	8.52	9.42	9.76
DPRNN [32]	2.65	21.9	0.012	11.34	9.41	8.63	9.79
Complex-MTASSNet [1]	28.18	1.9	0.019	12.57	9.86	8.42	10.28
EAD-Conformer [3]	26.09	2.1	0.005	13.37	11.41	10.56	11.78
ConformerNAS-A	16.30	1.5	0.004	13.65	11.71	10.89	12.08
ConformerNAS-B	20.98	1.8	0.005	13.61	11.90	11.11	12.21
ConformerNAS-C	36.67	2.8	0.006	13.67	12.00	11.23	12.30

Table 2: Comparison with other NAS methods.

Method	Params. (M)	SDRi (dB)			Avg.
		Speech	Music	Noise	
Random	16.96	13.54	11.04	10.14	11.57
Training-Free	16.60	13.62	11.35	10.76	11.91
Ours	16.30	13.65	11.71	10.89	12.08

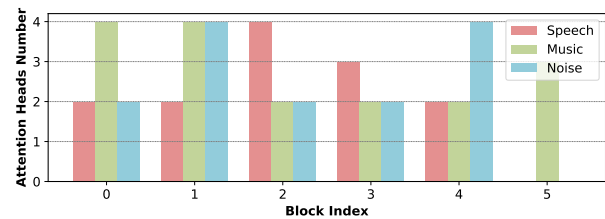
methods are shown in Table 1. Our searched architectures can customize the structure for the encoder and three decoders according to their characteristics of processing different signals, further improving the representation ability while removing the redundancy. The searched results prove this conclusion: ConformerNAS-A is our most lightweight model with the least FLOPs, which, however, outperformed other methods in all tracks. Our searched ConformerNAS-B and ConformerNAS-C have a little more FLOPs and can further push the state-of-the-art performance to a higher level.

3.3. Comparison with other NAS methods

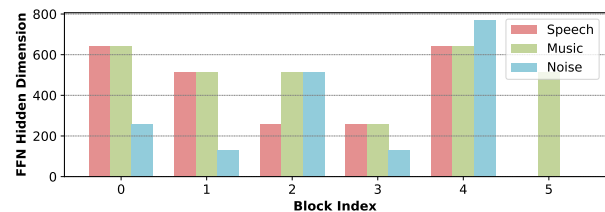
We search for a model with the similar size to our ConformerNAS-A using the random search [37] and a training-free search method [38] to provide an apples-to-apples comparison in Table 2. The randomly searched architecture performs worst, showing that finding a superb architecture in our enormous search space is non-trivial. Though the training-free method improves the searched performance with little search cost, it seems that the correlation between its training-free metric and the final performance is still not that good or reliable, and our ConformerNAS-A surpasses it with fewer parameters.

3.4. More analysis

We visualize the attention heads number and the FFN hidden dimension of each block for three decoders of ConformerNAS-C in Figure 4. As different signals have different characteristics, our searched architectures have distinct structures for each decoder. For example, the source music has high energy and is uniformly distributed among full frequency, the searched music decoder of ConformerNAS-C has one more block than its speech and noise decoders and has a larger FFN hidden di-



(a) Attention heads number vs. block index for three decoders.



(b) FFN hidden dimension vs. block index for three decoders.

Figure 4: Visualization of the attention heads and the FFN hidden dimensions of three decoders in ConformerNAS-C.

mension in most blocks, which achieves the highest SDRi in all three tracks, especially the music track, proving that such searched architecture is effective in improving the decoding performance of the music track, as shown in Table 1.

4. Conclusions

In this paper, we propose a comprehensive Conformer-based NAS search space and adopt a K-path search method to search for a customized architecture for the multi-task audio separation by a joint search in both macro (depth) and micro (dimension, head) levels. Our searched architectures achieve a new SOTA performance with fewer parameters and FLOPs, fully demonstrating the effectiveness of our method. However, our work only discovers the optimal structure of the standard Conformer block and doesn't invent new operations. In the future, we will further explore a more abundant search space, possibly with customized operations, to discover more effective architectures.

5. References

- [1] L. Zhang, C. Li, F. Deng, and X. Wang, “Multi-task audio source separation,” in *IEEE ASRU*, 2021.
- [2] D. Petermann, G. Wichern, Z.-Q. Wang, and J. L. Roux, “The cocktail fork problem: Three-stem audio separation for real-world soundtracks,” *arXiv preprint arXiv:2110.09958*, 2021.
- [3] C. Li, Y. Wang, F. Deng, Z. Zhang, X. Wang, and Z. Wang, “Ead-conformer: A conformer-based encoder-attention-decoder-network for multi-task audio source separation,” in *IEEE ICASSP*, 2022.
- [4] H. Wu, J. Xu, J. Wang, and M. Long, “Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting,” in *NeurIPS*, 2021.
- [5] S. Yu, T. Chen, J. Shen, H. Yuan, J. Tan, S. Yang, J. Liu, and Z. Wang, “Unified visual transformer compression,” in *ICLR*, 2022.
- [6] B. Zoph and Q. V. Le, “Neural architecture search with reinforcement learning,” in *ICLR*, 2017.
- [7] B. Baker, O. Gupta, N. Naik, and R. Raskar, “Designing neural network architectures using reinforcement learning,” in *ICLR*, 2017.
- [8] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, “Regularized evolution for image classifier architecture search,” in *AAAI*, 2019.
- [9] G. Bender, P.-J. Kindermans, B. Zoph, V. Vasudevan, and Q. Le, “Understanding and simplifying one-shot architecture search,” in *ICML*, 2018.
- [10] H. Liu, K. Simonyan, and Y. Yang, “Darts: Differentiable architecture search,” in *ICLR*, 2019.
- [11] X. Chu, X. Wang, B. Zhang, S. Lu, X. Wei, and J. Yan, “Darts: robustly stepping out of performance collapse without indicators,” in *ICLR*, 2021.
- [12] H. Pham, M. Y. Guan, B. Zoph, Q. V. Le, and J. Dean, “Efficient neural architecture search via parameter sharing,” in *ICML*, 2018.
- [13] J. Li, C. Liang, B. Zhang, Z. Wang, F. Xiang, and X. Chu, “Neural architecture search on acoustic scene classification,” in *Interspeech*, 2020.
- [14] N. W. Hasan, A. S. Saudi, M. I. Khalil, and H. M. Abbas, “E-darts: Enhanced differentiable architecture search for acoustic scene classification,” in *ICCES*, 2021.
- [15] —, “Automatically designing cnn architectures for acoustic scene classification,” in *ICCES*, 2021.
- [16] A. Mehrotra, A. G. C. Ramos, S. Bhattacharya, Ł. Dudziak, R. Vipplerla, T. Chau, M. S. Abdelfattah, S. Ishtiaq, and N. D. Lane, “Nas-bench-asr: Reproducible neural architecture search for speech recognition,” in *International Conference on Learning Representations*, 2021.
- [17] D. Rho, J. Park, and J. H. Ko, “Nas-vad: Neural architecture search for voice activity detection,” *arXiv preprint arXiv:2201.09032*, 2022.
- [18] W. Zhu, T. Kong, S. Lu, J. Li, D. Zhang, F. Deng, X. Wang, S. Yang, and J. Liu, “Speechnas: Towards better trade-off between latency and accuracy for large-scale speaker verification,” in *ASRU*, 2021.
- [19] E. Vincent, R. Gribonval, and C. Févotte, “Performance measurement in blind audio source separation,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.
- [20] Y. Wang, A. Narayanan, and D. Wang, “On training targets for supervised speech separation,” *IEEE/ACM transactions on audio, speech, and language processing*, vol. 22, no. 12, pp. 1849–1858, 2014.
- [21] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, “Conformer: Convolution-augmented transformer for speech recognition,” in *Interspeech*, 2020.
- [22] Y.-L. Liao, S. Karaman, and V. Sze, “Searching for efficient multi-stage vision transformers,” *arXiv preprint arXiv:2109.00642*, 2021.
- [23] Z. Guo, X. Zhang, H. Mu, W. Heng, Z. Liu, Y. Wei, and J. Sun, “Single path one-shot neural architecture search with uniform sampling,” in *ECCV*, 2020.
- [24] X. Chu, B. Zhang, and R. Xu, “Fairnas: Rethinking evaluation fairness of weight sharing neural architecture search,” in *ICCV*, 2021.
- [25] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: Nsga-ii,” *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [26] K. Tan and D. Wang, “Learning complex spectral mapping with gated convolutional recurrent networks for monaural speech enhancement,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 380–390, 2019.
- [27] A. Défossez, N. Usunier, L. Bottou, and F. Bach, “Music source separation in the waveform domain,” *arXiv preprint arXiv:1911.13254*, 2019.
- [28] N. Takahashi and Y. Mitsufuji, “D3net: Densely connected multidilated densenet for music source separation,” *arXiv preprint arXiv:2010.01733*, 2020.
- [29] Y. Luo and N. Mesgarani, “Conv-tasnet: Surpassing ideal time-frequency magnitude masking for speech separation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 8, pp. 1256–1266, 2019.
- [30] Y. Hu, Y. Liu, S. Lv, M. Xing, S. Zhang, Y. Fu, J. Wu, B. Zhang, and L. Xie, “Dccrn: Deep complex convolution recurrent network for phase-aware speech enhancement,” in *Interspeech*, 2020.
- [31] C. Subakan, M. Ravanelli, S. Cornell, M. Bronzi, and J. Zhong, “Attention is all you need in speech separation,” in *IEEE ICASSP*, 2021, pp. 21–25.
- [32] Y. Luo, Z. Chen, and T. Yoshioka, “Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation,” in *IEEE ICASSP*, 2020, pp. 46–50.
- [33] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, “Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline,” in *O-COCOSDA*, 2017, pp. 1–5.
- [34] T. Guo, C. Wen, D. Jiang, N. Luo, R. Zhang, S. Zhao, W. Li, C. Gong, W. Zou, K. Han *et al.*, “Didispeech: A large scale mandarin speech corpus,” in *IEEE ICASSP*, 2021, pp. 6968–6972.
- [35] N. Ono, Z. Rafii, D. Kitamura, N. Ito, and A. Liutkus, “The 2015 Signal Separation Evaluation Campaign,” in *International Conference on Latent Variable Analysis and Signal Separation*, 2015, pp. 186–190.
- [36] C. K. Reddy, H. Dubey, K. Koishida, A. Nair, V. Gopal, R. Cutler, S. Braun, H. Gamper, R. Aichner, and S. Srinivasan, “Interspeech 2021 deep noise suppression challenge,” *arXiv preprint arXiv:2101.01902*, 2021.
- [37] K. Yu, C. Sciuto, M. Jaggi, C. Musat, and M. Salzmann, “Evaluating the search phase of neural architecture search,” in *ICLR*, 2020.
- [38] W. Chen, X. Gong, and Z. Wang, “Neural architecture search on imagenet in four gpu hours: A theoretically inspired perspective,” in *ICLR*, 2021.