Interspeech 2024
1-5 September 2024, Kos, Greece

Streamlining Speech Enhancement DNNs: an Automated Pruning Method Based on Dependency Graph with Advanced Regularized Loss Strategies

Zugang Zhao[1], *Jinghong Zhang*[1], *Yonghui Liu*[2], *Jianbing Liu*[2], *Kai Niu*[1], *Zhiqiang He*[1]

[1]Key Laboratory of Universal Wireless Communications, Ministry of Education, Beijing University of Posts and Telecommunications(BUPT), Beijing, China
[2]Fanvil Link Technology Co., Ltd, Shenzhen, China

{zzgang, zjhong, niukai, hezq}@bupt.edu.cn, {james.liu, candy.liu}@fanvil.com

Abstract

In the burgeoning field of speech enhancement, the quest for high-performing deep neural networks(DNNs) often grapples with the challenge of increased computational demand and model size. This study unveils a novel structured pruning method that optimizes model via Dependency Graph, achieving automatic dimension reduction of network layers without manual settings of pruning ratios—a milestone not previously accomplished. Additionally, we propose a regularized loss strategy that adapts to variable scale sparsity, enhancing compression efficiency. Through extensive experiments, we demonstrate our method's ability to achieve substantial reductions in model size and computational costs while maintaining performance. Notably, Our findings also question the utility of grouping trick in linear layers, suggesting it may impede effective pruning. This research not only propels forward the capabilities of speech enhancement DNN compression, but also enriches the discourse on pruning methodology.

Index Terms: model compression, speech enhancement, structured pruning, reguarized loss

1. Introduction

Speech Enhancement (SE) aims to improve the clarity of target speech by attenuating background noise. Traditionally viewed through the lens of supervised learning[1], SE algorithms distinguish between noise and speech samples, generating masks that are applied to noisy audio in either the time domain [2], [3], [4], the time-frequency domain [5], [6], [7], or both [8], [9]. The advent of deep learning has significantly advanced SE, with deep learning based(DL-based) methods surpassing traditional signal processing techniques such as WebRTC Noise Suppression[10] in performance. However, the quest for superior performance has led to increased computational demands and larger model sizes, posing deployment challenges on edge devices like smartphones and hearing aids. Consequently, model compression technologies have emerged, serving as a critical link between academic research and practical engineering solutions.

Model compression techniques can be mainly categorized into size reduction and network quantization. Size reduction techniques like knowledge distillation[11], model pruning[12], and channel splitting[13] reduce model size, while network quantization increases computational efficiency by using fixed-point arithmetic[14]. This paper focuses on structured pruning[15], a size reduction strategy that eliminates redundancy at the column level in weight matrices, enhancing hardware acceleration compatibility. Substantial improvement reaches in recent studies. Tan et al.[16] developed a comprehensive model compression framework, integrating sparsity training, pruning, and quantization. Fang et al.'s DepGraph(DG)[17] algorithm automates pruning by generating dependency graph-based channel mappings. Chee et al.[18] optimized U-Net[19] by manually adjusting layer-specific pruning ratios, enhancing inference speed by seven times. However, two challenges remain: (1) conventional pruning techniques often focus on the concept of 'unreal' pruning that zeroing out weights without physically removing them, reducing model size but not computational load; (2) a uniform pruning ratio is always set across whole model and determined manually, potentially overlooking optimization opportunities. Addressing these issues, we introduce a novel structured pruning method that automatically identifies optimal pruning ratios for individual layers, improving model compression efficiency and overcoming the limitations of previous approaches. Our contributions are summarized as follows:

- Combine the automated nature of dependency graph generation and weight sensitivity analysis to achieve an effective structured pruning method for DL-based speech enhancement that enables actual dimension reduction of network layers without manual settings of pruning ratio, significantly compressing model size while keeping quality.
- Propose a innovative regularized loss combination that leverages DG to promote sparsity across a spectrum of scales within the weight matrix, from the finest (individual weights) to broader scales (channel-coupled weight groups), which leading to higher compression ratio of model.
- Raise doubts about the rationality of the widely-used grouping strategy[20] on the linear layer and the empirical analysis reveals that models subjected to grouping may exhibit lower pruning efficiencies and, in some cases, underperform compared to their non-grouped counterparts after pruning.

The rest of this paper is organized as follows. In Section 2, we describe our proposed approach in detail. In Section 3, we provide the experimental setup and depict the basic model to be pruned. Experimental results are presented and analyzed in Section 4. Section 5 concludes this paper.

2. Automated Structured Pruning Method

2.1. Overall Pruning Procedure

The proposed pruning methodology unfolds in four distinct phases as depicted in Figure 1: First, the dependency graph of model is generated to separate channel-coupled parameters detailed in section 2.2; Second, the model is initially trained with a specially designed regularized loss combination which is elaborated upon in Section 2.3; Third, the model undergoes pruning which is facilitated by an automatic search for compression ratio detailed in section 2.4; In the final phase, the pruned model is subject to fine-tuning, utilizing the same training setup but

10.21437/Interspeech.2024-785

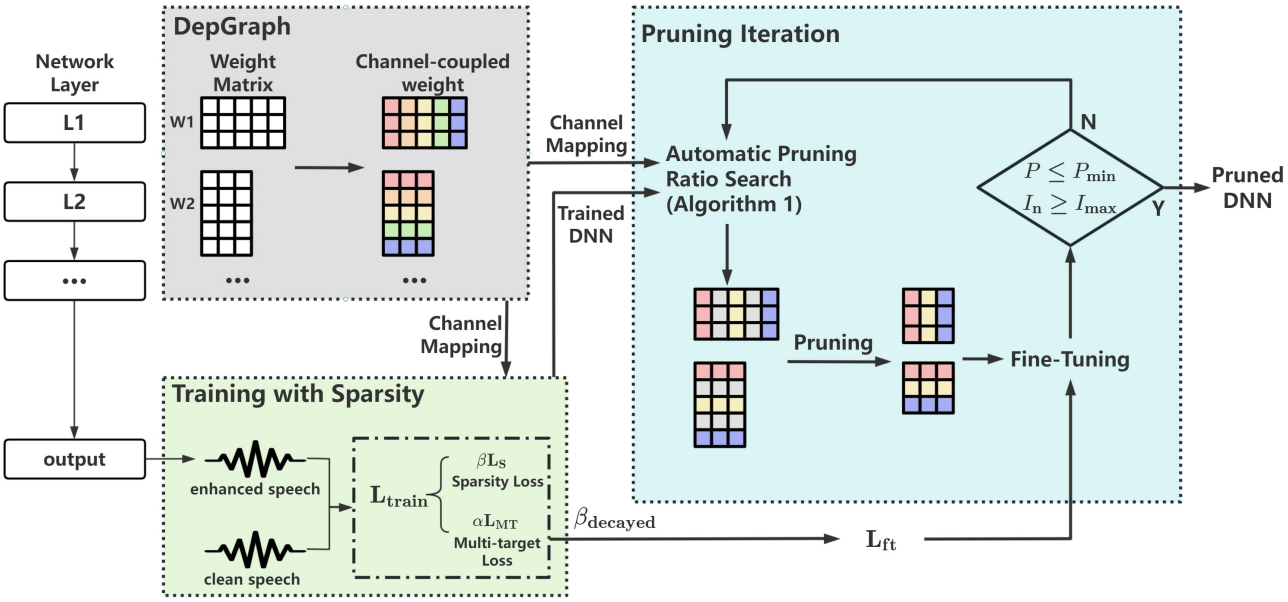

Figure 1: *Overall Pruning Procedure.*

excluding the decayed weights from the regularized loss component. The cycle of pruning and fine-tuning is considered a complete iteration of the pruning process and we repeat this iteration until meeting the criterion of early stopping that the reduction in the number of dimensions across the model becomes negligible after a iteration ($P \leq P_{\min}$) or the maximum allotted number of pruning iterations is reached($I_{\mathrm{n}} \geq I_{\max}$). $P_{\min}$ and $I_{\max}$ are set to 3 and 5, respectively.

2.2. Dependency Graph

DepGraph(DG)[17], or Dependency Graph, presents a novel method for understanding neural network structure through a dependency graph. This approach represents each network layer as a node and their interdependencies as edges, focusing on the impact of parameter pruning across layers. The graph is generated automatically by examining the architecture and data flow of the network. With this graph, DepGraph aligns the input and output channels of interconnected layers to ensure that pruning uniformly impacts channel-coupled parameters, thus maintaining the network's operational integrity. Despite its potential, DG struggles with layers involving direct dimensional adjustments such as permutation, flattening, or reshaping. These activities can result in output dimension mismatches during pruning, limiting applying DG on intricate models. Nonetheless, the ongoing development of DG's open-source code[21] continues to address these challenges.

2.3. Regularized Loss Strategies Based on DepGraph

For clarity, we define three grains of weight tensor that w means a signle weight, $\mathbf{c}$ means a column of the weight matrix and $\mathbf{g}$ means channel-coupled weights separated and packed by DG, identified by weights sharing the same color in Figure 1. A reguarized loss $\mathcal{L}_{\mathrm{g}}$ based on DG in the form of lasso penalty [22] is introduced for the $\mathbf{g}$ level sparsity, which can be formulated as:

$$\mathcal{L}_{\mathrm{g}} = \frac{1}{n(\mathcal{G})} \sum_{\mathbf{g} \in \mathcal{G}} n(p_{\mathbf{g}}) \|\mathbf{g}\|_2 \tag{1}$$

where $\mathcal{G}$ is the set of all groups of channel-coupled weights and $n(\cdot)$ is the counting function. $\|\cdot\|$ calculates the l_2 norm of tensor. with $\mathcal{L}_{\mathrm{g}}$, all weights within a channel group are simultaneously prompted either to be zero or not. Furthermore, the regularized loss in [16]:

$$\mathcal{L}_{\mathrm{SGL}} = \lambda_w \mathcal{L}_{\mathrm{w}} + \lambda_c \mathcal{L}_{\mathrm{c}} \tag{2}$$

$$= \frac{\lambda_w}{n(\mathcal{W})} \sum_{w \in \mathcal{W}} |w| + \frac{\lambda_c}{n(\mathcal{C})} \sum_{\mathbf{c} \in \mathcal{C}} \sqrt{p_{\mathbf{c}}} \|\mathbf{c}\|_2 \tag{3}$$

is introduced which further emphasizes sparsity on w and $\mathbf{c}$ scale in non-sparsity channel groups. Therefore, The proposed strategy of regularized loss is

$$\mathcal{L}_{\mathrm{S}} = \lambda_w \mathcal{L}_{\mathrm{w}} + \lambda_c \mathcal{L}_{\mathrm{c}} + \lambda_g \mathcal{L}_{\mathrm{g}} \tag{4}$$

where λ_w, λ_c and λ_g are the hyperparameter factor of each loss, respectively. Then $\mathcal{L}_{\mathrm{S}}$ is integrated with the multi-target loss $\mathcal{L}_{\mathrm{MT}}$ in [23] to train model:

$$\mathcal{L}_{\mathrm{train}} = \alpha \mathcal{L}_{\mathrm{MT}} + \beta \mathcal{L}_{\mathrm{S}} \tag{5}$$

where α and β are the predefined factor. $\mathcal{L}_{\mathrm{train}}$ guides the model towards enhanced performance while imposing sparsity across three levels of granularity which facilitates pruning. It should be noted that $\mathcal{L}_{\mathrm{S}}$ decays in fine-tuning process and the corresponding loss $\mathcal{L}_{\mathrm{ft}}$ can be revised as:

$$\mathcal{L}_{\mathrm{ft}} = \alpha \mathcal{L}_{\mathrm{MT}} + \beta \cdot \gamma^{n_{\mathrm{it}}} \mathcal{L}_{\mathrm{S}} \tag{6}$$

where $\gamma \in [0, 1]$ means the decaying factor and n_{it} denotes the sequence number of iterations.

2.4. Structured Pruning with Automatic Pruning Ratio Allocation Among Layers

DG plays a pivotal role in facilitating precise channel mapping across individual layers, thereby enabling 'real' sturctured pruning, i.e. actual channel reduction. However, an important challenge persists: the determination of pruning ratio. This

Table 1: *The Evaluation of models with Variable Compression Methods. '-p' represents pruning.*

Model	Para/M	MACs/M	PESQ	STOI	SISNR	CBAK	CSIG	COVL
noisy	-	-	1.97	0.920	8.42	2.44	3.34	2.63
DF2	2.00	219	2.51	0.922	14.0	3.00	3.40	2.93
DF2-p	**0.70**	**77.8**	2.53	0.922	14.3	3.05	3.40	2.93

Algorithm 1 Automatic Pruning Ratio Search

Input: (1) trained model $\mathrm{M_t}$; (2) dependency graph DG of $\mathrm{M_t}$; (3) validation set $\mathcal{V}$;
Output: pruning ratio σ_l for layer l in $\mathrm{M_t}$

for each layer l in $\mathrm{M_t}$ **do**
  Generate channel-grouped weights $\mathcal{G}_l$ based on DG and set $\sigma = 0\%$.
  while $\sigma \leq 95\%$ **do**
    Caculate l_1 norms of $g \in \mathcal{G}_l$ and find out the σ part of $\mathcal{G}_l$ with the smallest l_1 norms as $\mathcal{P}_l$;
    Derive pruned weights $\mathcal{G}_p$ and pruned model $\mathrm{M_p}$ by pruning $\mathcal{P}_l$ from $\mathcal{G}_l$;
    Caculate $D_{loss} = \mathcal{L}_{\mathrm{train}}(\mathcal{V} \mid \mathrm{M_p}) - \mathcal{L}_{\mathrm{train}}(\mathcal{V} \mid \mathrm{M_t})$;
    if $D_{loss} \geq \mathrm{Th}_D$ **then**
      break;
    else
      $\sigma = \sigma + 5\%$;
    end if
  end while
  Recover $\mathcal{G}_p$, $\mathrm{M_p}$ to original $\mathcal{G}_l$, $\mathrm{M_t}$.
  Return σ_l for layer l.
end for

issue is identified as a critical bottleneck in model compression efforts, as underscored in [18]. Drawing inspiration from the automated nature of the sensitivity analysis in [16], we introduce a novel methodology for automating the compression ratio search specially tailored for 'real' pruning, which can be depicted in Algorithm 1. The algorithm outlines an iterative approach to find the optimal pruning ratio for each layer in a trained deep neural network model. It leverages DG to guide the pruning process and iteratively tests different pruning ratios until an acceptable change in loss is observed, ensuring minimal impact on the model's performance. $\mathcal{L}_{\mathrm{train}}(\mathcal{V} \mid \mathrm{M_p})$ indicates evaluation with loss $\mathcal{L}_{\mathrm{train}}$ and The difference threshold Th_D is fixed to 0.1 in the subsequent experiments. The implementaion of the proposed algorithm can be found in https://github.com/zzgnb/Automated-Pruning/.

3. Experimental Setup

3.1. the Choice of Baseline

We selected DeepFilterNet2(DF2)[23] as the base model for pruning experiments due to its outstanding speech enhancement capabilities and relatively complex benchmark architectures like U-net, Codecs and CRN[24] in the lightweight DL-based SE area. The network topology of DF2, incorporating Fully Connected Layer (FCL), Convolutional Layer (Conv), and Gated Recurrent Unit (GRU), spans a broad spectrum of model designs such as DCCRN[6], Rnnoise[25], PercepNet[26], NSNet2[27], etc. We adapted the open-source DF2, making negligible changes for enhanced pruning, including splitting the original two-layer GRU into two single-layer units and canceling separation trick in Convs. The choice of DF2 was further motivated by its varied linear layers, a feature that is critically examined in Section 4.3. For training and fine-tuning, we set λ_w, λ_c, and λ_g in Equation 4 to 1e1, 2e-2, and 1e-5, and α, β, and γ in Equation 6 to 1, 1, and 0.9, respectively. The pruning of layers colored grey in Figure 2 is skipped because of the following direct dimensional operations.

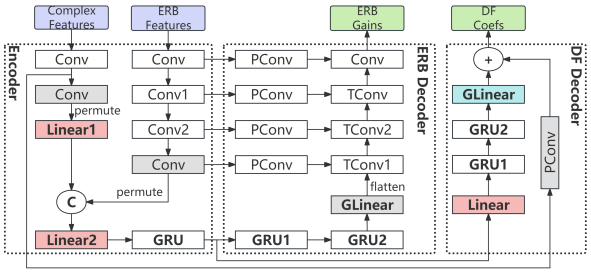

Figure 2: *The Architecture of DeepFilterNet2(DF2).*

3.2. Datasets

The model is trained and fine-tuned with identical dataset in [23]. For the evaluation phase, we leveraged the VoiceBank-Demand dataset[28]. This dataset is comprised of recordings from 28 native English speakers, presented across four distinct signal-to-noise ratios (SNRs) of 15, 10, 5, and 0 dB, culminating in a total of 11,572 utterances, all of which are sampled at 48KHz. Further details regarding the dataset's structure and characteristics can be found in the cited original publication.

3.3. Evaluation Metrics

We employ a suite of eight distinct metrics, embracing DNSMOS P.835[29] composed of Speech Distortion(SIG), Intrusiveness of Background Noise(BAK) and Overall Processed Speech Quality(OVL), Perceptual Evaluation of Speech Quality(PESQ)[30], Short-Time Objective Intelligibility(STOI)[31] and Scale Invariant Signal-to-Noise Ratio(SISNR)[32]. Notably, audio samples are resampled to 16KHz specifically for PESQ test. The Multiply Accumulate Operations (MACs) and the total number of parameters are applied to evaluate the computational cost and model size.

4. Experiments and Results

4.1. Optimization in Pruning Efficiency

The evaluation results, as captured in Table 1, underscore the efficacy of our proposed pruning method. The pruned version of DeepFilterNet2 (DF2) showcases a remarkable reduction in computational cost by 64.5% and model size by 65.0%, with negligible impact on its performance capabilities. Further insights provided in Table 4 reveal the nuanced adjustments in the dimensions of pruned layers, illustrating the method's ability to

Table 2: *The Evaluation of Models With Variable Sparsity. DF2-p(-S_{org}) represents the DF2 trained with original loss(Equation 2), which is the same as DF2-p in table 1. $-S_{new}$ represents training with the proposed integrated loss(Equation 4) while $-S_{non}$ denotes training without sparsity.*

Model	Para/M	MACs/M	PESQ	STOI	SISNR	CBAK	CSIG	COVL
DF2	2.00	219	2.51	0.922	14.0	3.00	3.40	2.93
DF2-p-S_{non}	0.93	103	2.54	0.922	14.2	3.03	3.44	2.97
DF2-p(-S_{org})	0.70	77.8	2.53	0.922	14.3	3.05	3.40	2.93
DF2-p-S_{new}	**0.68**	**76.0**	2.52	0.922	14.6	3.04	3.40	2.93

Table 3: *The Evaluation of Models With Variable Group Numbers. -ng indicates the number of groups for FCL in the model and (CR) denotes compression ratio.*

Model	oPara→pPara/M (CR)		oMACs→pMACs/M (CR)		PESQ	STOI	SISNR	CBAK	CSIG	COVL
DF2-p	2.00 → 0.70	**35.0%**	219 → 77.8	**35.5%**	2.53	0.922	14.3	3.05	3.40	2.93
DF2-2g-p	1.88 → 0.85	45.2%	207 → 94.3	45.6%	2.54	0.924	14.9	3.07	3.28	2.88
DF2-4g-p	1.83 → 0.81	44.3%	201 → 92.7	46.1%	2.54	0.923	15.0	3.10	3.35	2.92
DF2-8g-p	1.80 → 0.78	43.3%	199 → 88.2	44.3%	2.56	0.923	14.6	3.05	3.42	2.95
DF2-16g-p	1.78 → 0.75	42.1%	197 → 83.8	42.5%	2.49	0.920	15.3	3.07	3.33	2.88

Table 4: *Main Changes of the Input/Output Dimensions or Channels of Network Layers. Enc, Dec_{erb}, Dec_{df} Means Encoder, ERB Decoder and DF Decoder, respectively.*

Network Layer	$(\mathbf{In_{org}}, \mathbf{Out_{org}})$	$(\mathbf{In_{p}}, \mathbf{Out_{p}})$
Enc.Linear2	$(256, 256)$	$(250,\ 46\ )$
Enc.GRU	$(256, 256)$	$(\ 46\ , 256)$
Enc.Conv1	$(\ 16\ ,\ 16\ )$	$(\ 9\ ,\ 9\ )$
Enc.Conv2	$(\ 16\ ,\ 16\ )$	$(\ 9\ ,\ 10\ )$
Dec_{erb}.GRU1	$(256, 256)$	$(256,\ 37\ )$
Dec_{erb}.GRU2	$(256, 128)$	$(\ 37\ , 128)$
Dec_{erb}.TConv1	$(\ 16\ ,\ 16\ )$	$(\ 7\ ,\ 4\ )$
Dec_{erb}.TConv2	$(\ 16\ ,\ 16\ )$	$(\ 9\ ,\ 5\ )$
Dec_{df}.Linear	$(256, 256)$	$(256,\ 11\ )$
Dec_{df}.GRU1	$(256, 256)$	$(\ 11\ ,\ 24\ )$
Dec_{df}.GRU2	$(256, 256)$	$(\ 24\ , 232)$

dynamically optimize layer sizes—a level of adaptability and efficiency not achieved in prior works like [18]. These findings highlight the proposed method's enhanced flexibility and superiority in model optimization for speech enhancement tasks.

4.2. Advancements in Sparsity-Driven Loss

Table 2 presents the outcomes of training models with various regularized losses, underscoring the necessity of sparsity enforcement before pruning. Specifically, the comparison between DF2-p-S_{non} and DF2-p demonstrates the significant impact of sparsity on pruning. Moreover, DF2-p-S_{new} achieves higher compression ratios without compromising the model's functionality, showcasing the advantage of integrating sparsity at the channel-coupling level. Despite these advancements, the modest improvement prompts a reevaluation of our weight importance analysis method, currently utilizing the l_1 norm. This consideration reflects the ongoing debate in the academic field over identifying the most effective approach for evaluating weight significance, suggesting a need for further exploration and possibly the development of more refined techniques.

4.3. Reevaluating Grouping in Pruning Processes

Grouping in Fully Connected Layers (FCLs) is designed to reduce parameter redundancy, similar to pruning, by dividing input and output channels into groups for separate calculations. However, it may limits pruning efficiency as channels within the same group cannot be simultaneously eliminated. To assess grouping's impact on FCLs amenable to pruning, we categorized FCLs into three: (1) output FCLs (unprunable, blue in Figure 2); (2) hidden FCLs with direct dimensional operations (unprunable, grey in Figure 2); (3) hidden FCLs without above limitations (prunable, red in Figure 2). In our experiment, grouping was applied solely to the unprunable categories (1 and 2) with a fixed group size of 8, while category (3) explored various group configurations1, 2, 4, 8, and 16 groups. it's worth nothing that the removal of the shuffle operation[33] post-pruning, considering the infeasibility of parallel shuffle operations on channel-discontinuous tensors and the high computational cost of element-wise shuffling. Furthermore, the iterative pruning process naturally reassigns information among groups, realizing the role of shuffling to some extent. The experiment results presented in Table 3, revealing that models without grouping achieved higher compression ratios without quality loss, indicating that grouping may limit pruning potential and is unnecessary for prunable FCLs, negating the need to determine the optimal group number.

5. Conclusion

In this study, we introduced an automated pruning method for DNNs in speech enhancement, aimed at optimizing layer sizes for lower computational costs, complemented by a novel regularized loss combination. Evaluated on the DeepFilterNet2 model, our method achieved outstanding compression efficiency while retaining performance. Additionally, we questioned the benefit of grouping trick in linear layers, finding that it could impede the pruning process. Future research will focus on improving weight importance analysis techniques to further enhance the pruning efficacy of our method.

6. References

[1] D. Wang, "On ideal binary mask as the computational goal of auditory scene analysis," in *Speech separation by humans and machines*. Springer, 2005, pp. 181–197.

[2] E. Kim and H. Seo, "Se-conformer: Time-domain speech enhancement using conformer." in *Interspeech*, 2021, pp. 2736–2740.

[3] M. Strauss and B. Edler, "A flow-based neural network for time domain speech enhancement," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5754–5758.

[4] H. Shi, M. Mimura, L. Wang, J. Dang, and T. Kawahara, "Time-domain speech enhancement assisted by multi-resolution frequency encoder and decoder," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[5] J. Chen, Z. Wang, D. Tuo, Z. Wu, S. Kang, and H. Meng, "Fullsubnet+: Channel attention fullsubnet with complex spectrograms for speech enhancement," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7857–7861.

[6] 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," *arXiv preprint arXiv:2008.00264*, 2020.

[7] G. Zhang, L. Yu, C. Wang, and J. Wei, "Multi-scale temporal frequency convolutional network with axial attention for speech enhancement," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 9122–9126.

[8] Z. Zhang, S. Xu, X. Zhuang, Y. Qian, L. Zhou, and M. Wang, "Half-temporal and half-frequency attention u 2 net for speech signal improvement," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–2.

[9] S. Zhao and B. Ma, "D2former: A fully complex dual-path dual-decoder conformer network using joint complex masking and complex spectral mapping for monaural speech enhancement," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[10] Google, "Webrtc," https://webrtc.org.

[11] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[12] S. Vadera and S. Ameen, "Methods for pruning deep neural networks," *IEEE Access*, vol. 10, pp. 63 280–63 300, 2022.

[13] J. Yu and Y. Luo, "Efficient monaural speech enhancement with universal sample rate band-split rnn," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[14] P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh *et al.*, "Mixed precision training," *arXiv preprint arXiv:1710.03740*, 2017.

[15] Y. Sakai, Y. Eto, and Y. Teranishi, "Structured pruning for deep neural networks with adaptive pruning rate derivation based on connection sensitivity and loss function," *Journal of Advances in Information Technology*, vol. 1, 2022.

[16] K. Tan and D. Wang, "Towards model compression for deep learning based speech enhancement," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 29, pp. 1785–1794, 2021.

[17] G. Fang, X. Ma, M. Song, M. B. Mi, and X. Wang, "Depgraph: Towards any structural pruning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 16 091–16 101.

[18] J. Chee, S. Braun, V. Gopal, and R. Cutler, "Performance optimizations on u-net speech enhancement models," in *2022 IEEE 24th International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2022, pp. 1–6.

[19] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.

[20] Y. Ioannou, D. Robertson, R. Cipolla, and A. Criminisi, "Deep roots: Improving cnn efficiency with hierarchical filter groups," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1231–1240.

[21] "Torch-pruning," https://github.com/VainF/Torch-Pruning.

[22] J. Friedman, T. Hastie, and R. Tibshirani, "A note on the group lasso and a sparse group lasso," *arXiv preprint arXiv:1001.0736*, 2010.

[23] H. Schröter, A. Maier, A. Escalante-B, and T. Rosenkranz, "Deepfilternet2: Towards real-time speech enhancement on embedded devices for full-band audio," in *2022 International Workshop on Acoustic Signal Enhancement (IWAENC)*. IEEE, 2022, pp. 1–5.

[24] K. Tan and D. Wang, "A convolutional recurrent neural network for real-time speech enhancement." in *Interspeech*, vol. 2018, 2018, pp. 3229–3233.

[25] J.-M. Valin, "A hybrid dsp/deep learning approach to real-time full-band speech enhancement," in *2018 IEEE 20th international workshop on multimedia signal processing (MMSP)*. IEEE, 2018, pp. 1–5.

[26] J.-M. Valin, U. Isik, N. Phansalkar, R. Giri, K. Helwani, and A. Krishnaswamy, "A perceptually-motivated approach for low-complexity, real-time enhancement of fullband speech," *arXiv preprint arXiv:2008.04259*, 2020.

[27] S. Braun, H. Gamper, C. K. Reddy, and I. Tashev, "Towards efficient models for real-time deep noise suppression," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 656–660.

[28] C. Valentini-Botinhao, X. Wang, S. Takaki, and J. Yamagishi, "Speech enhancement for a noise-robust text-to-speech synthesis system using deep recurrent neural networks." in *Interspeech*, vol. 8, 2016, pp. 352–356.

[29] C. K. Reddy, V. Gopal, and R. Cutler, "Dnsmos p. 835: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 886–890.

[30] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs," in *2001 IEEE international conference on acoustics, speech, and signal processing. Proceedings (Cat. No. 01CH37221)*, vol. 2. IEEE, 2001, pp. 749–752.

[31] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech," in *2010 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2010, pp. 4214–4217.

[32] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 46–50.

[33] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6848–6856.