ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-6654-0540-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICASSP43922.2022.9746767

BLOOM-NET: BLOCKWISE OPTIMIZATION FOR MASKING NETWORKS TOWARD SCALABLE AND EFFICIENT SPEECH ENHANCEMENT

Sunwoo Kim, Minje Kim

Indiana University, Department of Intelligent Systems Engineering, Bloomington, IN, USA 47408

ABSTRACT

In this paper, we present a blockwise optimization method for masking-based networks (BLOOM-Net) for training scalable speech enhancement networks. Here, we design our network with a residual learning scheme and train the internal separator blocks sequentially to obtain a scalable masking-based deep neural network for speech enhancement. Its scalability lets it dynamically adjust the run-time complexity depending on the test time environment. To this end, we modularize our models in that they can flexibly accommodate varying needs for enhancement performance and constraints on the resources, incurring minimal memory or training overhead due to the added scalability. Our experiments on speech enhancement demonstrate that the proposed blockwise optimization method achieves the desired scalability with only a slight performance degradation compared to corresponding models trained end-to-end.

Index Terms— Speech Enhancement, ResNet, Model Compression, Scalability

1. INTRODUCTION

Deep learning-based supervised methods have dramatically boosted single-channel source separation performances in recent years. Typically, effective and dominating deep learning solutions operate by estimating masks, either on the time-frequency (TF) representations, such as the short-time Fourier transform (STFT) [1, 2, 3] or recent models that learn a *separator* module that applies masks in the latent feature space. The latter models have improved the state of the art as the learned feature space allows them to bypass limits imposed by TF-domain solutions (e.g., time-frequency resolution trade-off, using noisy phase or dealing with phase estimation, etc.) in addition to the advanced separator module's architecture. Various architectures have been proposed with each progressive model showing relative improvements: fully-convolutional Conv-TasNet [4] that initially popularized the time-domain approach, dual-path recurrent neural networks (DPRNN) that enabled long-term sequence modeling [5], and Transformer and Conformer-based models that overcome limitations from convolutional neural networks (CNN) and RNN based approaches (e.g., limitations of receptive fields and extensive recurrent connections) [6, 7, 8, 9].

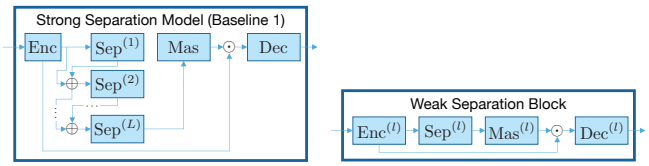
However, a major drawback of aforementioned deep learning solutions is the complexity of models. Heavy memory occupancy and especially their exorbitant computational cost makes them impractical for deployment onto resource-constrained devices.

Model compression methods offer effective solutions to this problem by reducing complexity of neural network architectures while minimizing their drop in generalization performance. There are various model compression methods such as quantizing model parameters using low-bit resolution fixed-point representations and/or pruning less important network components [10, 11] , simplifying convolutional operations [12], grouping RNN's intermediate tensor representations into smaller blocks [13], multi-resolution features via successive downsampling and resampling [14], distilling knowledge from a larger network to improve the performance of its corresponding compressed model [15, 16], etc. These compressed models are typically designed to minimize the inference complexity targeting the low-resource environment, thus not being able to scale up to challenging separation problems. Eventually, in order for a legacy system to be scalable, a range of versions need to be retained in a device, increasing the total spatial complexity.

We argue that a scalable and efficient system must cover a wide resource-related diversity in edge computing via an *adaptive model architecture* rather than simply enumerating various model architectures. The scalable, thus adaptive systems can be commonly found in coding applications. In [17], the cross-module residual learning scheme enabled greedy module-wise neural codec learning, where a deep autoencoder is trained to model the residual signal that its preceding autoencoder fails to model. The system can preserve the order of relative importance of the participating autoencoder modules, that gives scalability to the system, i.e., the first part of the bitstream is more important than the rest. SoundStream audio codec features bitrate scalability, allowing the codec to adapt to the network conditions that can vary while transmitting signals, too [18]. However, these models are specifically for signal compression, seeking scalability within their resulting bitstreams, thus not suitable for other applications. Meanwhile, the once-for-all (OFA) scheme provides a general-purpose adaptive training mechanism that learns multiple compressed variants of a model via a single training task [19]. However, it does not provide a single architecture that scales to different test environments freely as we propose.

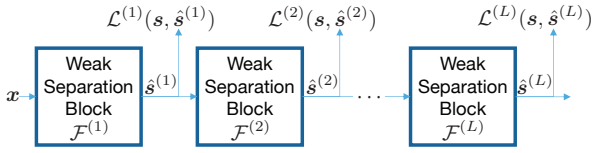
Likewise, we envision a scalable speech enhancement model that changes its operation modes, ranging from an energy-efficient version to a performance-boosted one. In doing so, instead of preparing each different model in the device, a scalable model provides a flexible structure that adjusts its performance and complexity per resource constraint. To this end, we propose a scalable time-domain architecture for speech enhancement: BLOckwise Optimization for Masking networks (BLOOM-Net). It is a greedy residual learning strategy to train individual blocks sequentially. Since each block is trained to improve the previous block's speech enhancement result further, the deployment can optionally choose to use only the first few blocks depending on the available computational resources. Although it has been known that a sequence of two heterogeneous speech enhancement processes is effective [20, 21], our approach differs from that literature in that (a) ours focuses on the architectural innovation rather than a concatenation of two heterogeneous models (b) the proposed model eventually achieves the scalability in the fea-

This material is based upon work supported in part by the National Science Foundation under Grant Numbers 1909509 and 2046963.

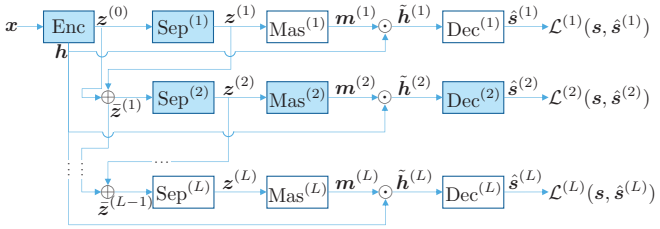


(a) (b)

Fig. 1: (a) The baseline separation model (b) a weak separation block



(a) The time-domain blockwise optimization model (Baseline 2).



(b) The proposed BLOOM-Net architecture.

Fig. 2: The scalable speech enhancement models.

ture space rather than in the raw signal domain. BLOOM-Net shows competitive performance compared to their end-to-end counterparts, while providing the additional advantage that its modularized blocks are easy to attach or detach for scalability.

2. METHODOLOGIES

2.1. Baseline 1: The Time-Domain Separation Model

Our baseline source separation model adopts the common structure found in time-domain source separation networks that consists of an encoder, separator, masker, and decoder as shown in Fig. 1a. First, the input $\boldsymbol{x} \in \mathbb{R}^T$ with T samples in the time domain is fed to the encoder, which is a function that transforms the input into a latent representation: Enc : $\boldsymbol{x} \rightarrow \boldsymbol{h} \in \mathbb{R}^D$. We posit that it is a small neural network module, e.g., a 1D convolutional layer followed by a nonlinear activation function. Then, the latent representation is fed into the separator module: Sep : $\boldsymbol{h} \rightarrow \boldsymbol{z} \in \mathbb{R}^K$, whose output $\boldsymbol{z}$ is used as input to the mask estimation module Mas : $\boldsymbol{z} \rightarrow \boldsymbol{m} \in \mathbb{R}^D$. Finally, the mask is applied to the encoded mixture representation $\boldsymbol{h}$ to retrieve the source-specific estimate of the latent representation $\tilde{\boldsymbol{h}}$ (e.g., for the speech source), which is then decoded back to the time-domain estimate of the target source, Dec : $\tilde{\boldsymbol{h}} \rightarrow \hat{\boldsymbol{s}} \in \mathbb{R}^T$.

In the time-domain separation models, it is common to employ a large separator module with repeating structures indexed by l. It is also popular to merge the input and output of each block as input to the next one, performing residual learning as proposed in ResNet [22]. These blocks are learned altogether in the state-of-the-art models as well as in Baseline 1, limiting the models' scalability.

2.2. Baseline 2: Time-Domain Blockwise Optimization

To build up our proposed BLOOM-Net model, we begin with a naïve concatenation approach as our second baseline. Fig. 1b shows a *weak* source separation block as a compromised version of Baseline 1 in Fig. 1a. Although it is with only one ResNet block for the separator module, it is still a legitimate stand-alone separation model. We assume that there are L such weak separation blocks, each of which enhances its previous block's results as shown in Fig. 2a. Let $\mathcal{F}^{(l)}(\cdot)$ be the l-th weak separation block. It performs speech enhancement on $\hat{\boldsymbol{s}}^{(l-1)}$, an output of the $(l-1)$-th block: $\mathcal{F}^{(l)} : \hat{\boldsymbol{s}}^{(l-1)} \rightarrow \hat{\boldsymbol{s}}^{(l)}$. Note that the notion of a "weak" separation block comes from the boosting methods that incrementally add weak learners [23].

The second baseline provides a mechanism to serialize the speech enhancement models with an order of significance. If the $(l-1)$-th weak separation block leaves room for improvement, i.e., when the loss $\mathcal{L}^{(l-1)}(\boldsymbol{s}, \hat{\boldsymbol{s}}^{(l-1)})$ is not sufficiently small, the next block $\mathcal{F}^{(l)}(\hat{\boldsymbol{s}}^{(l-1)})$ focuses on that sample and tries to improve. During the test time, suppose the device can afford only up to ℓ blocks. Since each of the L blocks is sequentially trained with its own reconstruction loss $\mathcal{L}^{(l)}(\boldsymbol{s}, \hat{\boldsymbol{s}}^{(l)})$, scalability is achieved by performing the inference on only the first $l \leq \ell$ weak separation blocks—the result is still a legitimate source estimate. In contrast, Baseline 1 would need to redundantly prepare all the scaled variants of the model in order to achieve the desired scalability.

2.3. BLOOM-Net: Blockwise Optimization in the Latent Space

Baseline 2 exhibits a redundancy issue. During the test time, the chosen block should execute all four submodules to deliver a time-domain signal to its successive block. Instead, we propose a blockwise optimization scheme that works in the latent space, so that the serialization is done among the separator submodules. In that way, the system can avoid unnecessary masking, decoding, and encoding operations that repeat at every weak separation block in Baseline 2.

Fig. 2b describes the BLOOM-Net architecture, where $\text{Sep}^{(l)}$ performs residual learning in the sense that it relays the sum of its output $\boldsymbol{z}^{(l)}$ and the input $\bar{\boldsymbol{z}}^{(l-1)}$ to the next separator block:

$$\bar{\boldsymbol{z}}^{(l)} = \boldsymbol{z}^{(l)} + \bar{\boldsymbol{z}}^{(l-1)}, \text{ where } \boldsymbol{z}^{(l)} = \text{Sep}^{(l)}\big(\bar{\boldsymbol{z}}^{(l-1)}\big). \quad (1)$$

Note that the encoder's output $\boldsymbol{h} = \boldsymbol{z}^{(0)}$ is the input to $\text{Sep}^{(1)}$. We also induce the input to $\text{Sep}^{(L)}$ recursively: $\bar{\boldsymbol{z}}^{(L-1)} = \sum_{l=0}^{L-1} \boldsymbol{z}^{(l)}$.

BLOOM-Net performs block-specific masking and decoding just to compute the block-specific error, while the residual connections are defined among the latent variables $\boldsymbol{z}^{(l)}$. The l-th masker and decoder works on the separator output $\boldsymbol{z}^{(l)}$ as follows:

$$\hat{\boldsymbol{s}}^{(l)} = \text{Dec}^{(l)}\big(\tilde{\boldsymbol{h}}^{(l)}\big),\ \tilde{\boldsymbol{h}}^{(l)} = \boldsymbol{m}^{(l)} \odot \boldsymbol{h}^{(l)},\ \boldsymbol{m}^{(l)} = \text{Mas}^{(l)}\big(\boldsymbol{z}^{(l)}\big). \quad (2)$$

The blockwise output $\hat{\boldsymbol{s}}^{(l)}$ is then compared with the ground-truth source $\boldsymbol{s}$ to compute the blockwise loss $\mathcal{L}^{(l)}(\boldsymbol{s}, \hat{\boldsymbol{s}}^{(l)})$, ensuring the intermediate output is usable. Note that Enc is shared and reused for all the sequence of blocks such that all blocks learn to estimate denoising masks on the same latent representation $\boldsymbol{h}$. Meanwhile, $\text{Mas}^{(l)}$ and $\text{Dec}^{(l)}$ modules are block-specific and no longer updated once the l-th block is trained.

During the test time, the actual inference involves Enc, $\text{Sep}^{(l)}$ where $1 \leq l \leq \ell$, $\text{Mas}^{(l)}$, and $\text{Dec}^{(l)}$, which are represented as shaded blocks in Fig. 2b. Compared to Baseline 2, BLOOM-Net saves the cost of $\ell - 1$ encoding, decoding, and masking operations. Although these blocks are lightweight, removing them improves not only the

computational efficiency but also feature learning. The direct residual learning path in the latent feature space allows BLOOM-Net to learn a hierarchy of latent representations, while Baseline 2 is limited to concatenating shallow representations.

Fine-tuning can further refine a fully trained BLOOM-Net. While BLOOM-Net provides the desired scalability via blockwise optimization, its greedy nature prevents the model from learning from the full picture, thus underperforming the theoretical upper bound. To this end, we propose to fine-tune BLOOM-Net, where all modules in all L blocks are updated using the combination of all loss functions, i.e., $\sum_{l=1}^{L} \mathcal{L}^{(l)}(\boldsymbol{s}, \hat{\boldsymbol{s}}^{(l)})$. This comprehensive fine-tuning significantly improves BLOOM-Net, making its performance near the theoretical upper bound.

3. EXPERIMENTAL SETUP

3.1. Datasets

During training, we used clean speech samples from the Librispeech corpus [24] and noise recordings from the MUSAN dataset [25]. We used `train-clean-100` and `dev-clean` subsets from Librispeech for training and validation respectively. We split MUSAN's `free-sound` subset at 80:20 ratio into training and validation partitions. For testing, we used unseen speech samples from Librispeech's `test-clean` and noise from MUSAN's `sound-bible`. Audio files are loaded at 16 kHz sampling rate and standardized to have a unit-variance. Noise samples are scaled to random input SNR levels uniformly chosen between -5 and 10 dB and added to speech signals to obtain noisy mixtures.

3.2. Architecture and Training Details

To investigate its application to time-domain audio separation networks [4, 5, 6, 7, 26], our ResNet-based models are implemented as a simplified form of Conv-TasNet [4]. We use the same encoder and decoder design as in [4] with the same hyperparameters. Each separator module is defined as a residual block using 1-D convolutional layers similar to those in [4], but without dilation and intermediate skip-connections. Each residual block consists of a 1×1 convolution operation followed by a depthwise convolution operation, with parametric ReLU (PReLU) [27] and global layer normalization (gLN) added after each convolution operation. We used the ConvTasNet implementation available in Asteroid [28] and tuned the hyperparameters to construct our time-domain ResNet models.

We used the negative scale-invariant signal to distortion ratio (SI-SDR) as the loss function [29], defined as

$$\mathcal{L}(\boldsymbol{s}, \hat{\boldsymbol{s}}) = -\text{SI-SDR}(\boldsymbol{s}, \hat{\boldsymbol{s}}) = -10\log_{10}\left(\frac{||\alpha\boldsymbol{s}||^2}{||\alpha\boldsymbol{s} - \hat{\boldsymbol{s}}||^2}\right) \quad (3)$$

where $\alpha = \frac{\hat{\boldsymbol{s}}^\top \boldsymbol{s}}{||\boldsymbol{s}||^2}$ is a scaling factor. We train all models on 1-second long segments using a mini-batch size of 64. Adam is used as the optimizer with learning rate initialized to 1×10^{-4}. Early stopping is applied and models with the lowest validation losses are used for final evaluation on the test mixtures.

3.3. Training Configurations

The experiments are on two types of baselines and our proposed BLOOM-Net method. We additionally examine the impact of fine-tuning on BLOOM-Net.

Table 1: SI-SDR improvements of the competing models. Evaluations are with respect to the number of blocks ℓ chosen for inference.

Method	$\ell = 1$	$\ell = 2$	$\ell = 3$	$\ell = 4$	$\ell = 5$	$\ell = 6$
Baseline 1 - Full	-0.60	-0.86	-0.12	0.40	0.97	8.89
Baseline 1 - Int.	4.55	6.83	7.65	8.30	8.58	8.89
Baseline 2	4.55	5.01	5.17	5.25	5.25	5.25
BLOOM	4.55	6.13	6.92	7.40	7.61	7.75
BLOOM-FT	4.74	6.55	7.44	8.14	8.51	8.72

• *Baseline 1 - Full*: The time-domain separation model (Sec. 2.1) trained in a conventional end-to-end manner. It employs $L = 6$ separator blocks from the beginning and learns all of them together. Although there are $L = 6$ separators, removing one of them to reduce the complexity will cause a complete break down as the model is not scalable.

• *Baseline 1 - Int.*: We also learn $L = 6$ different versions of Baseline 1, each of which is an intermediate version, containing up to $L = \{1, \ldots, 6\}$ separators, respectively, e.g., when $L = 3$ there are three separators that are trained altogether. Each of these models is the upper bound of its corresponding BLOOM-Net with a matching separator number.

• *Baseline 2*: The time-domain weak separation blocks (Sec. 2.2). Individual blocks are trained as a stand-alone separation model one after another to promote scalability.

• *BLOOM*: BLOOM-Net in its basic setup (Sec. 2.3).

• *BLOOM-FT*: BLOOM-Net, fine-tuned to minimize the combination of losses from all blocks. It overcomes the suboptimal performance caused by *BLOOM*'s greedy training.

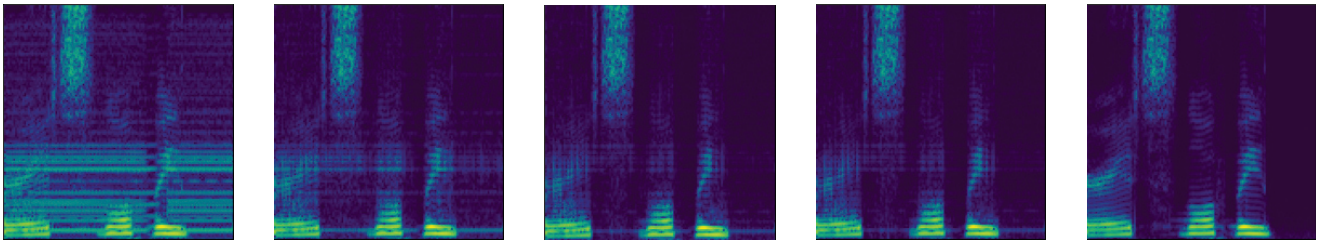
4. EXPERIMENTAL RESULTS AND DISCUSSION

Table 1 presents the results of the competing systems. We first compare BLOOM-Net with Baseline 1. At a glance, the model trained under *Baseline 1 - Full* achieves the highest performance when it fully utilizes all 6 separator blocks. However, the model's scalability is limited, as it is evidently unable to perform denoising when less than 6 separator blocks are used (columns from $\ell = 1$ to 5). This behavior is expected since all 6 modules are trained altogether in the conventional end-to-end fashion—its intermediate separator outputs $\boldsymbol{z}^{(l)}$ are not suitable for the shared masker module to compute the mask from, except for the final separator output $\boldsymbol{z}^{(6)}$.

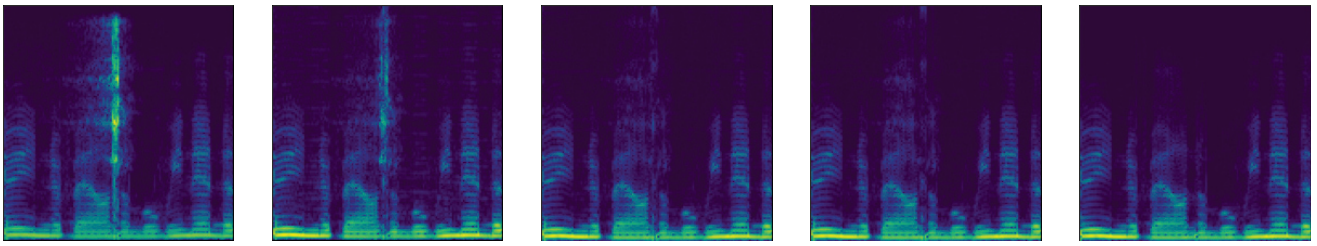
Hence, it is unavoidable for Baseline 1 to train multiple versions of different block configurations to scale to various application and hardware requirements, which is the *Baseline 1 - Int.* setup. Since in this set up there are totally six different end-to-end models, each of which specializes in each choice of ℓ, they form the performance upper bound. However, the system's total spatial complexity is the sum of all six versions, which is not the most efficient option.

BLOOM-Net, on the other hand, exhibits desired scalability. Instead of completely failing, *BLOOM* shows decent performance at $\ell = 5$ case, which is only a 0.1 dB drop from its $\ell = 6$ setup. Its performance apparently drops as the model complexity decreases more. In addition, *BLOOM*'s most powerful setup $\ell = 6$ is suboptimal compared to Baseline 1.

BLOOM-FT addresses this issue by fine-tuning the entire modules via the sum of all blockwise loss functions. Compared against the oracle end-to-end Baseline 1 results (*Baseline 1 - Int.)*, the fine-tuned results usually show less than 0.3 dB drop in all cases. Hence, we claim that the properly fine-tuned BLOOM-Net is almost comparable to the traditional end-to-end models while it provides unprece-



(a) $\boldsymbol{x}_1$ (SI-SDR: 7.63 dB) (b) $\hat{\boldsymbol{s}}_1^{(2)}$ (SI-SDR: 12.12 dB) (c) $\hat{\boldsymbol{s}}_1^{(4)}$ (SI-SDR: 13.13 dB) (d) $\hat{\boldsymbol{s}}_1^{(6)}$ (SI-SDR: 13.29 dB) (e) $\boldsymbol{s}_1$



(f) $\boldsymbol{x}_2$ (SI-SDR: 3.05 dB) (g) $\hat{\boldsymbol{s}}_2^{(2)}$ (SI-SDR: 14.22 dB) (h) $\hat{\boldsymbol{s}}_2^{(4)}$ (SI-SDR: 19.93 dB) (i) $\hat{\boldsymbol{s}}_2^{(6)}$ (SI-SDR: 19.98 dB) (j) $\boldsymbol{s}_2$

Fig. 3: Denoising output samples from intermediate blocks. Each row represents a different example. The columns represent noisy mixture, estimated reconstructions from l-th blocks, and the corresponding ground-truth target clean speech.

dented scalability and spatial efficiency.

Next, we draw attention towards the time-domain blockwise optimization method, *Baseline 2*. Compared to BLOOM-Net that operates in the latent domain, *Baseline 2* sequentially feeds time-domain inputs to the next stand-alone module. This incurs an overhead of learning the feature transformation and its inverse operation, constraining each block to only learn a shallow latent representation; thus, the improvement by adding more weak separation blocks is only minimal. It showcases the merit of the proposed BLOOM-Net algorithm that performs blockwise optimization in the latent space.

Fig. 3 shows two denoising examples where the different model complexity choices affect the quality of the output. The deeper the inputs are processed within the BLOOM-Net, the higher the denoising quality. The depth of the network can be decided based on the requirements of the deployed environment or users' preference. This demonstrates BLOOM-Net's scalability feature that offers flexible quality-complexity tradeoff to adjust to the test environment.

Finally, Table 2 shows the inference-time computational complexity of *Baseline 1 - Int.* and BLOOM-Net in terms of number of parameters and multiply-accumulate (MAC) operations. Note that *BLOOM* and *BLOOM-FT* are equivalent in this context. First, there is no difference in computational complexity (MACs), as the active modules during inference are the same. However, BLOOM-Net exhibits a significant advantage in terms of spatial complexity when we assume a scalable model. For example, if the device can afford up to $\ell = 2$, the baseline has to prepare two different versions with $L = 1$ and $L = 2$ for the best performance in both energy-efficient and performance-boosted use cases. In doing so, although these two versions' sizes are $0.28M$ and $0.42M$ parameters, respectively, their sum amounts to $0.70M$. Likewise, in order for Baseline 1 to be scalable, it exponentially accumulates spatial complexity as ℓ grows. Conversely, BLOOM-Net manages this scalability issue more carefully: it increases the model size just by the amount of a single separator block and its insignificantly small block-specific masker and decoder modules (about 0.21M parameters). Hence, for example, our scalable BLOOM-Net with five separators (1.13M) is smaller than the baseline that covers three complexity profiles (1.26M).

Table 2: Computational requirements of time-domain ResNet models trained under Baseline 1 and our proposed BLOOM-Net method. The number of parameters reported encompasses the entire model parameters needed to implement the scalable model. MACs are computed given 1-second inputs.

	Method	$\ell = 1$	$\ell = 2$	$\ell = 3$	$\ell = 4$	$\ell = 5$	$\ell = 6$
MACs (G)	Baseline 1	0.53	0.80	1.07	1.33	1.60	1.87
	BLOOM						
Params (M)	Baseline 1	0.28	0.70	1.26	1.95	2.78	3.64
	BLOOM	0.28	0.49	0.71	0.92	1.13	1.34

5. CONCLUSION

In this study, we introduced BLOOM-Net, a novel algorithm for scalable speech denoising. We postulated that scalable implementation of a deep learning-based speech enhancement system is critical to handle various resource-related test conditions that a device faces. While traditional end-to-end time-domain source separation models have shown advanced separation performance, we claimed that such a system cannot provide the desired scalability. Our BLOOM-Net is with a carefully designed residual learning scheme that performs blockwise optimization to improve the model performance in an incremental way. Since the blockwise optimization was on each individual separator module that greedily contributes to the model performance, BLOOM-Net achieves the scalablility—the device can freely choose from the multiple profiles based on its resource constraint. In doing so, the enhancement quality is reasonably associated with the model complexity. Source codes are available at https://saige.sice.indiana.edu/research-projects/bloom-net.

6. REFERENCES

[1] A. Narayanan and D. L. Wang, "Ideal ratio mask estimation using deep neural networks for robust speech recognition," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2013.

[2] D. S. Williamson, Y. Wang, and D. Wang, "Complex ratio masking for monaural speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 3, pp. 483–492, March 2016.

[3] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep Clustering: Discriminative Embeddings for Segmentation and Separation," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2016.

[4] 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.

[5] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-path RNN: efficient long sequence modeling for time-domain single-channel speech separation," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2020.

[6] J. Chen, Q. Mao, and D. Liu, "Dual-path transformer network: Direct context-aware modeling for end-to-end monaural speech separation," in *Proc. of the Annual Conference of the International Speech Communication Association (Interspeech)*, 2020.

[7] C. Subakan, M. Ravanelli, S. Cornell, M. Bronzi, and J. Zhong, "Attention is all you need in speech separation," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2021, pp. 21–25.

[8] S. Chen *et al.*, "Continuous speech separation with conformer," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2021.

[9] Y. Koizumi *et al.*, "DF-Conformer: Integrated architecture of Conv-TasNet and Conformer using linear complexity self-attention for speech enhancement," in *Proc. of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2021.

[10] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," in *Proc. of the International Conference on Learning Representations (ICLR)*, 2016.

[11] S. Kim, M. Maity, and M. Kim, "Incremental binarization on recurrent neural networks for single-channel source separation," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2019.

[12] A. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[13] Y. Luo, C. Han, and N. Mesgarani, "Group communication with context codec for lightweight source separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1752–1761, 2021.

[14] E. Tzinis, Z. Wang, and P. Smaragdis, "Sudo rm -rf: Efficient Networks for Universal Audio Source Separation," in *Proc. of the IEEE Workshop on Machine Learning for Signal Processing (MLSP)*, 2020, pp. 1–6.

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

[16] S. Kim and M. Kim, "Test-time adaptation toward personalized speech enhancement: Zero-shot learning with knowledge distillation," in *Proc. of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2021.

[17] K. Zhen, J. Sung, M. S. Lee, S. Beack, and M. Kim, "Cascaded cross-module residual learning towards lightweight end-to-end speech coding," in *Proc. of the Annual Conference of the International Speech Communication Association (Interspeech)*, 2019.

[18] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "Soundstream: An end-to-end neural audio codec," *arXiv preprint arXiv:2107.03312*, 2021.

[19] H. Cai, C. Gan, T. Wang, Z. Zhang, and S. Han, "Once-for-all: Train one network and specialize it for efficient deployment," in *Proc. of the International Conference on Learning Representations (ICLR)*, 2019.

[20] Y. Zhao, Z.-Q. Wang, and D. Wang, "Two-stage deep learning for noisy-reverberant speech enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 1, pp. 53–62, 2018.

[21] X. Hao *et al.*, "Masking and inpainting: A two-stage speech enhancement approach for low SNR and non-stationary noise," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2020.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[23] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *icml*, vol. 96. Citeseer, 1996, pp. 148–156.

[24] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[25] D. Snyder, G. Chen, and D. Povey, "MUSAN: A Music, Speech, and Noise Corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[26] Y. Luo and N. Mesgarani, "Tasnet: time-domain audio separation network for real-time, single-channel speech separation," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2018.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[28] M. Pariente *et al.*, "Asteroid: The PyTorch-Based Audio Source Separation Toolkit for Researchers," in *Proc. of the Annual Conference of the International Speech Communication Association (Interspeech)*, 2020.

[29] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "SDR – half-baked or well done?" in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2019.