Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.042816

ARTICLE

Joint On-Demand Pruning and Online Distillation in Automatic Speech Recognition Language Model Optimization

Soonshin Seo[1,2] and Ji-Hwan Kim[2,*]

[1]Clova Speech, Naver Corporation, Seongnam, 13561, Korea

[2]Department of Computer Science and Engineering, Sogang University, Seoul, 04107, Korea

*Corresponding Author: Ji-Hwan Kim. Email: kimjihwan@sogang.ac.kr

Received: 13 June 2023 Accepted: 13 October 2023 Published: 26 December 2023

ABSTRACT

Automatic speech recognition (ASR) systems have emerged as indispensable tools across a wide spectrum of applications, ranging from transcription services to voice-activated assistants. To enhance the performance of these systems, it is important to deploy efficient models capable of adapting to diverse deployment conditions. In recent years, on-demand pruning methods have obtained significant attention within the ASR domain due to their adaptability in various deployment scenarios. However, these methods often confront substantial trade-offs, particularly in terms of unstable accuracy when reducing the model size. To address challenges, this study introduces two crucial empirical findings. Firstly, it proposes the incorporation of an online distillation mechanism during on-demand pruning training, which holds the promise of maintaining more consistent accuracy levels. Secondly, it proposes the utilization of the Mogrifier long short-term memory (LSTM) language model (LM), an advanced iteration of the conventional LSTM LM, as an effective alternative for pruning targets within the ASR framework. Through rigorous experimentation on the ASR system, employing the Mogrifier LSTM LM and training it using the suggested joint on-demand pruning and online distillation method, this study provides compelling evidence. The results exhibit that the proposed methods significantly outperform a benchmark model trained solely with on-demand pruning methods. Impressively, the proposed strategic configuration successfully reduces the parameter count by approximately 39%, all the while minimizing trade-offs.

KEYWORDS

Automatic speech recognition; neural language model; Mogrifier; long short-term memory; pruning; distillation; efficient deployment; optimization; joint training

1 Introduction

1.1 Advancements in ASR

Recent advancements in end-to-end automatic speech recognition (ASR), which transforms spoken language into written text, have led to the development of cutting-edge models that are suitable for industrial applications [1–5].

Acoustic models (AMs) play a crucial role in ASR systems, translating acoustic signals into phonetic units. Their advancements have been foundational in constructing more robust end-to-end ASR systems. Main techniques in this field include connectionist temporal classification (CTC) [6], a training principle tailored for sequence-to-sequence tasks. It does not require a precise match between the input and output, which is extremely valuable in ASR, where it can often be challenging to establish a clear alignment between audio segments and their corresponding text representations. The recurrent neural network transducer [7] offers an improved approach to predicting output sequences of various lengths, capitalizing on the strengths of recurrent neural networks (RNNs). Another method is the attention-based encoder-decoder [8,9], where an encoder digests the input sequence, and a decoder produces the output. An attention mechanism pinpoints parts of the input sequence most pertinent to each output step.

Furthermore, there have been advancements in decoding methods for end-to-end ASR models, which are responsible for generating the final transcription based on model predictions. These methods vary from simple greedy decoding, where the model selects the most likely word or phoneme at each step, to more complex approaches like frame-synchronous beam search [10–12]. The latter simultaneously considers multiple potential transcriptions to determine the most probable one. One noteworthy development is the incorporation of shallow fusion with autoregressive neural language models (LMs). In this approach, the ASR model's predictions are combined with a separate LM to refine the transcription. The use of beam search in this fusion process has proven to be crucial in enhancing ASR performance [1,4,5], as depicted in Fig. 1.

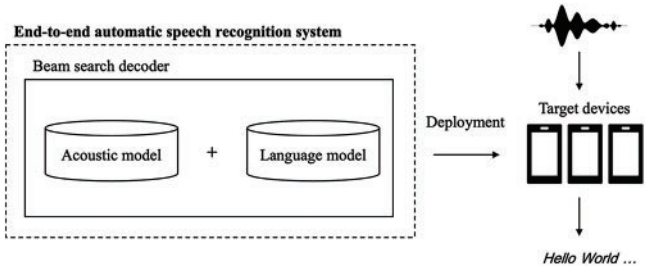


Figure 1: Overview of the end-to-end automatic speech recognition system in this study

Fig. 1 provides a detailed illustration of the end-to-end ASR system used in this study. This system incorporates the shallow fusion method, combining the AMs and LMs, and employs the beam search method for decoding. One of the central objectives of this system is its flexibility, making it compatible with a wide range of devices, from powerful computers to less resource-intensive gadgets. This adaptable approach is designed to address the challenges arising from varying computational capabilities and device specifications, making it an attractive solution for a multitude of real-world applications.

In the domain of LMs, RNNs, such as long short-term memory (LSTM) [13–15], have played a crucial role in enhancing the learning of long-term dependencies and mitigating challenges related to vanishing or exploding gradients. Nevertheless, Transformers [16] have taken these strengths further by leveraging self-attention mechanisms to facilitate inter-word interactions, thereby enabling more intricate dependencies, and enhancing contextual understanding. However, conventional Transformers have grappled with the problem of context fragmentation. To tackle this issue, Transformer-XL [17] not only resolves the context fragmentation problem but also captures longer-range dependencies

through a recurrent mechanism and a unique positional encoding scheme, resulting in substantial performance enhancements.

1.2 Challenges in ASR Deployment

Notwithstanding these remarkable strides, the development and effective deployment of ASR systems for resource-constrained environments, such as mobile devices, persist as formidable challenges [18–20]. ASR systems often operate in scenarios where high-demand computational resources may not be readily accessible, compelling the utilization of low-complexity models. Additionally, there exists a compelling need for these systems to be adaptable across a spectrum of devices concurrently. This necessity underscores the requirement for a transferable ASR model capable of deployment across a spectrum ranging from high-end devices to entry-level ones, each characterized by distinct specifications and varying maximum available model sizes. While, in theory, it might be conceivable to train a distinct model for each device under the assumption of limitless resources. However, such an approach would ultimately be highly inefficient.

Several prior investigations have delved into the realm of efficient deployments where a solitary model operates across diverse conditions. One widely adopted approach to address this challenge is the use of slimmable networks [21–23], along with on-demand pruning [24–27]. These methods are part of the broader spectrum of pruning strategies [28–32]. Specialized training methods, such as the "random" [24] or "sandwich rule" [22], enable on-demand pruning to be trained in a single pass. Subsequently, the trained model can be dynamically applied during inference, streamlining the deployment of models with varying sizes based on a single adaptable base model. Despite its adaptability and efficiency, inherent challenges persist, including a trade-off between the number of model parameters and corresponding performance [21–23]. Consequently, when reducing the model parameters, ensuring consistent performance becomes a non-trivial task.

To further enhance the adaptability of ASR systems in resource-limited environments, LSTM-based models have demonstrated promise due to their recurrent architecture [33], enabling efficient processing of variable-length sequences with fewer parameters and reduced computational demands compared to Transformers. Notably, the Mogrifier LSTM [34], an advanced iteration of the conventional LSTM, augments the model's expressive capabilities without introducing substantial complexity. Its performance has exhibited great potential, frequently surpassing that of vanilla LSTMs and Transformer-XLs, all while preserving a comparable computational complexity [33,34].

1.3 Contributions

This study proposes the incorporation of an online distillation mechanism during the on-demand pruning training phase. This is devised to ensure the stability of model accuracy, primarily by minimizing the disparity between the pruned model's predictions and the original model's output probabilities, consequently yielding more robust results. Additionally, this study introduces the Mogrifier LSTM LM as the target for pruning within the ASR framework. Given the characteristics of the Mogrifier LSTM, it emerges as an ideal candidate for an on-demand pruning strategy tailored for deployment in resource-constrained environments.

To substantiate the efficacy of the proposed methods, this study conducts a series of experiments on the ASR system employing the Mogrifier LSTM LM, which is trained using the suggested joint on-demand pruning and online distillation approach. The experimental outcomes, as observed on the LibriSpeech test-other subset, clearly demonstrate that our proposed methods surpass a baseline model trained solely through on-demand pruning techniques. Moreover, our research verifies that the

proposed configuration can reduce the model parameters by approximately 39%. This reduction is accomplished with minimal trade-offs, underscoring the potential efficiency of our approach in terms of computational resources, all the while maintaining the model's performance at a satisfactory level.

2 Related Works

In this section, this study delves into prior research studies that bear relevance to our proposed approach of joint on-demand pruning and online distillation. Section 2.1 offers an overview of the evolution of LMs in the context of ASR. Sections 2.2 and 2.3 expound upon the various pruning strategies and introduce the underlying distillation mechanisms, respectively. Finally, in Section 2.4, we scrutinize previous works that have sought to integrate these two methods.

2.1 LMs in End-to-End ASR

LMs have been employed in tandem with AMs within end-to-end ASR systems, with the overarching goal of enhancing transcription accuracy. LMs play a crucial role in aiding ASR systems in generating the most likely linguistic hypotheses for sentences. In practical application, LMs are constructed as probabilistic models, which may encompass n-gram models or neural network LMs. Various types of LMs can be seamlessly integrated into end-to-end ASR configurations. The most basic form of integration is commonly referred to as LM shallow fusion.

Conventional n-gram LMs operate under the Markov assumption but face challenges, including difficulties in recognizing unfamiliar n-grams and limitations associated with small n values [35]. To overcome these issues, deep neural networks (DNNs) are introduced, utilizing complex vector representations to represent words [36,37]. However, as the length of sequences increased, DNNs exhibited constraints. This led to the emergence of RNNs, capable of handling longer sequences [38]. The bidirectional variant of RNNs, which considers both preceding and succeeding contexts, demonstrated superior performance, albeit encountering issues such as gradient vanishing or explosion [39–41]. The introduction of LSTM cells addressed these gradient problems [13–15,42]. While LSTM models outperformed traditional RNNs, they did require lengthier training times [43].

The introduction of attention mechanisms marked developments in the field, leading to the emergence of the Transformer model. This model incorporates positional encoding, further establishing its importance [16,44–46]. Within the family of Transformer-based models, BERT stands out for its excellence in natural language processing, thanks to its bidirectional encoding. On the other hand, GPT-2, known for its multi-layer decoding, excels in language modeling tasks [47,48]. Transformer-XL has its unique advantages, particularly in specific applications, due to its ability to retain past states [17].

2.2 Pruning Strategies

2.2.1 Conventional Pruning

Pruning strategies play a crucial role in the realm of neural network optimization, offering a suite of techniques aimed at removing redundant connections through iterative fine-tuning processes [28–32]. These refined models effectively reduce the number of parameters while preserving the accuracy of the original model, making them indispensable in resource-constrained scenarios.

Han et al. [28] introduced a method to tackle the demanding computational and memory requirements frequently associated with neural networks. Their achievement lies in the identification and subsequent pruning of redundant network connections. This approach results in a significant

reduction in parameters while upholding the model's accuracy, effectively streamlining neural network architectures, and mitigating the resource-intensive nature of deep learning.

Wu et al. [29] contributed to this field by introducing BlockDrop, an ingenious strategy characterized by dynamic layer selection during inference. This approach effectively reduces computational overhead while concurrently preserving high accuracy levels. The substantial acceleration of inference up to 36% attained through their approach underscores its importance in the context of neural network optimization.

Liu et al. [30] challenged traditional notions surrounding network pruning. Their proposition shifts the focus from preserving specific important weights to emphasizing the significance of the pruned architecture itself in achieving an efficient final model. This fresh perspective redefines the approach to network pruning, illuminating alternative avenues for attaining computational efficiency.

Collectively, these previous studies underscore the dynamic landscape of pruning strategies in neural network optimization, providing solutions to the challenges posed by burgeoning computational demands across diverse environments. Nonetheless, the adoption of these approaches in specific settings necessitates thoughtful consideration of associated expenses, particularly in the context of the expanding array of devices and applications.

2.2.2 On-Demand Pruning

Conversely, on-demand pruning can be dynamically applied during inference, obviating the necessity for further fine-tuning. Through the utilization of on-demand pruning, the creation of large yet lightweight models concurrently become feasible. Furthermore, it offers versatile deployment possibilities, adaptable to diverse environments.

Several training methods have been proposed to effectively implement on-demand pruning. One such approach is commonly referred to as "random" [24–27], which targets specific layers or modules within a single neural network for pruning. These targets are selected randomly during the training phase. For instance, Huang et al. [24] introduced stochastic depth to train deep convolutional networks, involving the random omission of layers during training. This enables the training of shorter networks while using deeper networks during testing. Fan et al. [25] introduced LayerDrop, a form of structured dropout, to regulate over-parameterized transformer networks. It allows efficient pruning during inference by selecting sub-networks from a larger network without requiring additional fine-tuning. Lee et al. [26] presented a training and pruning method for ASR models that employ intermediate CTC and stochastic depth to reduce model depth at runtime without the need for extra fine-tuning. Vyas et al. [27] proposed stochastic compression for compute-efficient Wav2vec 2.0 models, which incorporates a variable squeeze factor and query and key-value pooling mechanisms for compression. The stochastically pre-trained model can be fine-tuned for specific configurations, resulting in substantial computational savings.

While random sampling is relatively simple to implement, its intrinsic randomness can be challenging to control, and it does not consistently guarantee accuracy compared to the non-pruned model. Another method known as the "sandwich rule" [21–23] provides an alternative approach. Yu et al. [21] devised a technique for training a single neural network at various widths, which they termed Slimmable neural networks. This approach dynamically adjusts the network width during runtime, striking a balance between efficiency and accuracy. Building upon the concept of the Slimmable network, they introduce universally Slimmable networks [22] based on the sandwich rule. This allows these networks to operate at any chosen width. The sandwich rule is devised to train sub-networks with varying configurations within a single neural network, and the losses from these

sub-networks are aggregated and back-propagated simultaneously. This method proves particularly effective in optimizing neural networks when working within defined lower and upper bounds.

Moreover, Li et al. [23] introduced the dynamic Slimmable network, a dynamic network slimming technique designed to enhance hardware efficiency. This network shows considerable performance improvements over other model compression techniques, achieving reduced computation and real-world acceleration, while only minimally impacting accuracy in image classification.

2.3 Distillation Mechanisms

2.3.1 Teacher-Student Distillation

Distillation mechanisms have found extensive application in enhancing the performance and efficiency of neural networks [49–52]. In the conventional framework, a larger "teacher" network imparts its structured knowledge to a smaller "student" network, enabling the student network to leverage this guidance and achieve better performance compared to training from scratch.

In a more specific context, Hinton et al. [49] introduced an approach to consolidate the knowledge from a variety of models into a single model, greatly streamlining the deployment process. They introduce a type of ensemble consisting of one or more comprehensive models alongside several specialist models designed to distinguish fine-grained classes, which are often challenging for the full models to differentiate.

Sun et al. [50] presented a patient knowledge distillation approach that efficiently compresses a large-scale teacher model into a more compact student model without compromising effectiveness. The student model patiently learns from multiple layers of the teacher model, employing two strategies: learning from the last few layers and learning from every few layers. This approach leads to improved training efficiency and enhances performance across various natural language processing tasks by leveraging the rich information present in the hidden layers of the teacher model.

In a related study, Jiao et al. [51] introduced a Transformer distillation approach explicitly tailored for the knowledge distillation of Transformer-based models. They introduced a two-stage learning framework for TinyBERT, where Transformer distillation is performed during both the pretraining and task-specific learning phases. Nevertheless, these approaches necessitate a fixed network of pre-trained teachers, which can be costly and restrict the flow of knowledge from the teacher network to the student network in a one-way manner.

2.3.2 Online Distillation

Conversely, online distillation mechanisms streamline the training process by treating all networks as student networks, allowing them to exchange knowledge within a single stage [53–55]. Lan et al. [53] introduced an online distillation technique known as an on-the-fly native ensemble. This method eliminates the need for a complex two-phase training procedure associated with offline distillation methods. Instead, it simultaneously trains a single multi-branch network while dynamically constructing a potent teacher model on the fly.

Furthermore, in research by Zhang et al. [54], a deep mutual learning strategy was proposed. Instead of the traditional one-way knowledge transfer from a teacher to a student, this approach promotes a collaborative learning environment where an ensemble of students teaches each other throughout the training process. This eliminates the requirement for a powerful pre-trained teacher network. Notably, this collaborative learning approach enhances the performance of various network architectures and surpasses distillation methods reliant on a more capable yet static teacher.

Expanding upon the principles of the preceding two approaches, Guo et al. [55] introduced an effective online knowledge distillation method that leverages collaborative learning. In contrast to traditional two-stage knowledge distillation methods that adhere to a unidirectional learning paradigm, this method fosters a single-stage training environment where knowledge is shared among students during collaborative training. Experimental results underscore the consistent improvement in the performance of all student models across diverse datasets, and it proves effective in knowledge transfer to multiple tasks, including object detection and semantic segmentation.

2.4 Integration of Pruning and Distillation Mechanisms

Pruning strategies and distillation mechanisms have traditionally been employed as separate techniques, but contemporary research has shifted towards their integration. The objective is to optimize the benefits of both methods, streamlining their combination to reduce computational overhead and enhance overall efficiency [56]. Distillation has emerged as a valuable approach to address the inherent loss of precision associated with pruning. However, it is essential to note that applying distillation after the pruning process, rather than concurrently with pruning during training, might potentially lead to suboptimal model performance [57].

The recent advancement in the field of ASR is the integration of both pruning and distillation methods. Initially, various methods have been explored for applying pruning to ASR based on self-supervised learning (SSL) [58–60]. Building on these foundational studies, Peng et al. [60] highlighted the limitations of using knowledge distillation in SSL. A significant drawback they pointed out is the necessity for manual design and maintaining a fixed architecture for the student model during training. Such an approach is dependent on prior knowledge and might not always achieve optimal performance. Addressing these challenges, they introduced task-agnostic compression methods for voice SSL, drawing inspiration from task-specific structural pruning. By merging distillation and pruning techniques, their methods showcased superiority over methods based solely on distillation across multiple tasks.

The preceding studies indicate that an integrated approach involving pruning and distillation reveals the complementary nature of these two strategies when training neural networks for diverse tasks. This serves as a basis for extending the same concept to the online mechanism and LM task domains addressed in this study.

3 Joint On-Demand Pruning and Online Distillation for ASR LM

As discussed in Section 2, on-demand pruning offers efficient deployment and adaptability to various conditions, including diverse devices. However, it can encounter accuracy fluctuations when employed to compress models into smaller sizes. In such cases, distillation mechanisms can serve as a valuable complement. In Section 3, this study delves into the proposed joint on-demand pruning and online distillation approach, outlining its application in an online fashion for ASR LMs. Furthermore, this study analyzes and validates the efficacy of the Mogrifier LSTM LM as a potential target for pruning within the ASR task.

In Section 3.1, this study introduces the Mogrifier LSTM as the selected LM for on-demand pruning, and Section 3.2 details the baseline, delving into the utilization of the random and sandwich rule as methods for on-demand pruning. Finally, in Section 3.3, this study describes the proposed method that merges the sandwich rule with online distillation mechanisms.

3.1 Target LM for On-Demand Pruning

3.1.1 LSTM

LSTM is a type of RNN architecture. RNNs are neural networks specifically designed to process sequences, like time series data or natural language. The LSTM is introduced to address the limitations of the basic RNN, primarily the problem of long-term dependencies where the RNN struggles to remember information from earlier in the sequence. The LSTM introduces the concept of a cell state, a kind of memory that flows through the network and can be added to, removed from, or read using three distinct gates.

Specifically, for a given input vector sequence $\mathbf{x}_1, \ldots, \mathbf{x}_T \in \mathbb{R}^m$, a vanilla LSTM [16] processes input vector x_t sequentially to produce an output vector $\mathbf{h}_t \in \mathbb{R}^n$ and a cell state $\mathbf{c}_t \in \mathbb{R}^n$. These two outputs are then used to produce the next output $\mathbf{h}_{t+1}$ for the next input $\mathbf{x}_{t+1}$. The function of the LSTM layer is denoted as:

$$\mathbf{c}_t, \mathbf{h}_t = \text{LSTM}(\mathbf{x}_t, \mathbf{c}_{t-1}, \mathbf{h}_{t-1}). \tag{1}$$

The LSTM layer has three gates: forget (**f**), input (**i**), and output gates (**o**). The output of the gates is used to produce $\mathbf{c}_t$ and $\mathbf{h}_t$. The forget gate determines which portions of the cell state should be discarded. The input gate integrates new information into the cell state. Subsequently, the output gate ascertains the segments of the cell state to be read and emitted. The gates employ sigmoid activation functions that yield values in the range of 0 to 1, facilitating their decision-making processes. The operation proceeds as follows:

$$\mathbf{f}_t = \sigma\left(\mathbf{W}_f \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f\right), \tag{2}$$

$$\mathbf{i}_t = \sigma\left(\mathbf{W}_i \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i\right), \tag{3}$$

$$\mathbf{o}_t = \sigma\left(\mathbf{W}_o \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o\right), \tag{4}$$

$$\hat{\mathbf{c}}_t = \tanh\left(\mathbf{W}_c \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_c\right), \tag{5}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \hat{\mathbf{c}}_t, \tag{6}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh\left(\mathbf{c}_t\right), \tag{7}$$

where σ is the sigmoid non-linearity function, $\odot$ is the element-wise product operation, and $\mathbf{W}_*$ and $\mathbf{b}_*$ are the weight matrix and bias vector.

3.1.2 Mogrifier LSTM

With the advancement of deep learning, there have been endeavors to enhance the capabilities of the conventional LSTM. One notable development in this regard is the Mogrifier LSTM [32]. This variant introduces additional gates that are designed to iteratively refine the input to the LSTM. These refinements aim to optimize the LSTM's interaction with its inputs, ultimately improving its learning effectiveness.

During each iteration within these gates, there is a selective update of either the input sequence or the preceding hidden state, depending on the value of the other. This mechanism aligns with the fundamental principles of LSTM's gating mechanisms. The iterative rounds facilitate a nuanced interplay between the input and the hidden state, thereby enhancing the model's proficiency in recognizing intricate patterns within the data.

Subsequent studies on the Mogrifier LSTM [31,32,61] consistently demonstrate that various Mogrifier variants outperform both the traditional LSTM and Transformer models in language modeling tasks.

Specifically, in the Mogrifier LSTM, the input of the vanilla LSTM, $\mathbf{x}_{t-1}$, and $\mathbf{h}_{t-1}$ are replaced with the output of a special layer, referred to as the "MogrifierGate". Accordingly, Eq. (1) is changed to:

$$\mathbf{x}_{t,r}, \mathbf{h}_{t-1,r} = \text{MogrifierGate}\left(\mathbf{x}_t,\ \mathbf{h}_{t-1}\right), \tag{8}$$

$$\mathbf{c}_t, \mathbf{h}_t = \text{LSTM}\left(\mathbf{x}_{t,r},\ \mathbf{c}_{t-1},\ \mathbf{h}_{t-1,r}\right). \tag{9}$$

The structure of the MogrifierGate is based on a series of "rounds", denoted as r where r is greater than or equal to 1. For each round, a gate vector is obtained using either $\mathbf{x}_t$ or $\mathbf{h}_{t-1}$. Then, the other component ($\mathbf{x}_t$, if $\mathbf{h}_{t-1}$ is used to obtain the gate vector, and vice versa) is updated using its previous value and the gate vector. Formally, for the i-th round, if i is odd:

$$\mathbf{x}_{t,i} = 2\sigma\left(\mathbf{W}_i^Q \cdot \mathbf{h}_{t-1,i-1}\right) \odot \mathbf{x}_{t,i-2}, \tag{10}$$

and if i is even:

$$\mathbf{h}_{t-1,i} = 2\sigma\left(\mathbf{W}_i^R \cdot \mathbf{x}_{t,i-1}\right) \odot \mathbf{h}_{t-1,i-2}, \tag{11}$$

where σ is a sigmoid non-linearity, $\odot$ is the element-wise product, and $\mathbf{W}_i^*$ refers to the weight matrix for the i-th round.

3.1.3 The Impact of Pruning Mogrifier Rounds

Furthermore, to explain the impact of pruning on the components of the Mogrifier LSTM, this study analyzes their inference times and presents the results in Fig. 2. Specifically, this study compares the inference time required for Eqs. (8) and (9), which differ in the number of rounds they involve. This analysis reveals that for a Mogrifier LSTM layer with 8 rounds, calculating these rounds (as per Eq. (8)) takes 2.8 α times longer than the computation of the vanilla LSTM (as per Eq. (9)), where α represents the inference time of the latter. This suggests that when conducting a feed-forward operation with a Mogrifier LSTM comprising eight rounds, it occupies 74% of the total inference time. However, the inference time needed for Eq. (8) can be significantly reduced to 0.76 α when the number of rounds is decreased to 1. This implies that effective pruning of rounds with minimal or no performance degradation can reduce the necessary inference time by more than half (from $\alpha + 2.8\alpha$ to $\alpha + 0.76\alpha$).

3.2 Baseline: Sandwich Rule for On-Demand Pruning

Previously, this study discussed the benefits of pruning rounds within the Mogrifier LSTM in the context of computational efficiency. In this section, this study establishes a baseline for our research. The baseline approach entails the utilization of the sandwich rule for on-demand pruning, with a specific focus on pruning Mogrifier rounds. This method initiates with a random pruning mechanism.

3.2.1 Random

The random strategy involves generating a random value s, where $1 \leq s \leq r$, for each iteration, and then using only the s-th MogrifierGate to feed-forward the input for that iteration. This strategy is straightforward to implement and has proven effective for on-demand pruning in prior research

[22–25]. Formally, the random strategy can be described as follows:

$$s \sim \text{Uniform}\,(1, r), \tag{12}$$

$$\mathbf{x}^b_{t,s}, \mathbf{h}^b_{t-1,s} = \text{MogrifierGate}_s\left(\mathbf{x}^b_t, \mathbf{h}^b_{t-1}\right), \tag{13}$$

$$\mathbf{c}^b_t, \mathbf{h}^b_t = \text{LSTM}\left(\mathbf{x}^b_{t,s}, \mathbf{c}^b_{t-1}, \mathbf{h}^b_{t-1,s}\right). \tag{14}$$

Here, MogrifierGate_s represents the sub-gate of MogrifierGate for the s-th round, and b represents the mini-batch. It should be noted that s is shared among all timesteps ($1 \leq t \leq T$) and mini batches. After training, this study evaluates the model using a fixed round.

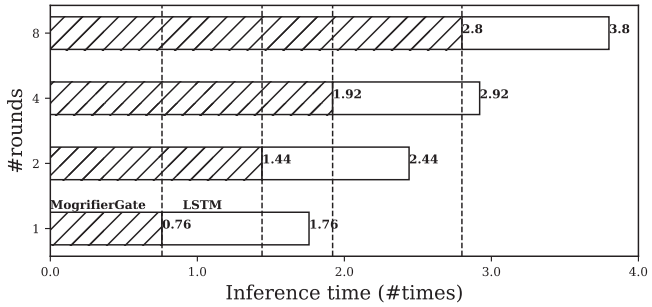


Figure 2: An analysis on the inference time of the two components of a Mogrifier long short-term memory (LSTM) layer using one CPU (Intel Xeon Gold 5120): MogrifierGate and LSTM layer

3.2.2 Sandwich Rule

Random selection has been reported in practice that the smallest sub-model ($s = 1$) and the largest sub-model ($s = r$) tend to have reduced accuracy compared to the typical s round Mogrifier LSTM. To address this limitation, the two sub-models are required to be trained more cautiously. Yu et al. [20] proposed a sandwich rule that is employed in universally Slimmable networks, and this study adopts it for the training of MogrifierGate pruning. The sandwich rule aims to mitigate the limitation on the accuracy of the smallest and largest sub-models by additionally weighting them during training.

This study considers the use of multiple objective functions $\mathcal{L}_s$, where $\mathcal{L}_s$ is the objective function of the model using the s round Mogrifier sub-gate, MogrifierGate_s, obtained from the r round of MogrifierGate. This study considers multiple values of $\mathcal{L}_s$ concurrently using the sandwich rule. Formally, the loss of sandwich rule can be addressed as:

$$s_1 = 1, \tag{15}$$

$$s_2 = r, \tag{16}$$

$$s_3, \ldots, s_k \sim \textit{Uniform}\,(2, r-1), \tag{17}$$

$$\mathcal{L} := \frac{1}{k}\sum\nolimits_{i=1}^{k} \mathcal{L}_{s_i}, \tag{18}$$

where k is the hyperparameter of the sandwich rule, representing the number of losses per training iteration (here, $k \geq 2$) and $\mathcal{L}$ is used for training the model.

One drawback of the sandwich rule is the increase in computational cost for each mini-batch, as it requires separate computations for each $\mathcal{L}_{s_i}$. Nevertheless, through experimental observations, this study notices that the sandwich rule significantly enhances the final accuracy of both the smallest and largest models, surpassing the performance of the random method.

3.3 Proposed Method: Sandwich Rule with Online Distillation Mechanism

In this section, this study introduces an approach that combines on-demand pruning and an online distillation mechanism customized for ASR LMs. Specifically, the proposed method builds upon the foundational concept of the sandwich rule, which is explained in Section 3.2. However, this study focuses on the fact that the utility of the sandwich rule may exhibit instability when the demand for more compressed models becomes paramount. To confront this challenge, this study proposes the integration of an online distillation mechanism in tandem with on-demand pruning during the training phase. This distillation mechanism takes inspiration from a previous method, namely collaborative learning-based online distillation, which is discussed in detail in Section 2.3. The primary advantage of this approach resides in its capacity to consistently accrue additional information through the ensemble of soft targets generated by all student networks throughout the training process [55].

The proposed method can be conceptualized as an extension of the sandwich rule, as illustrated in Fig. 3. Assuming the existence of k sub-models predicated on the sandwich rule, we denote the logits of the i-th sub-model as $\mathbf{z}_i$, while the ensemble logits generated from all sub-models are designated as $\mathbf{z}_{ensemble}$. This can be formally expressed as follows:

$$\mathbf{z}_{ensemble} = \frac{1}{k}\sum_{i=1}^{k} \mathbf{z}_i. \tag{19}$$

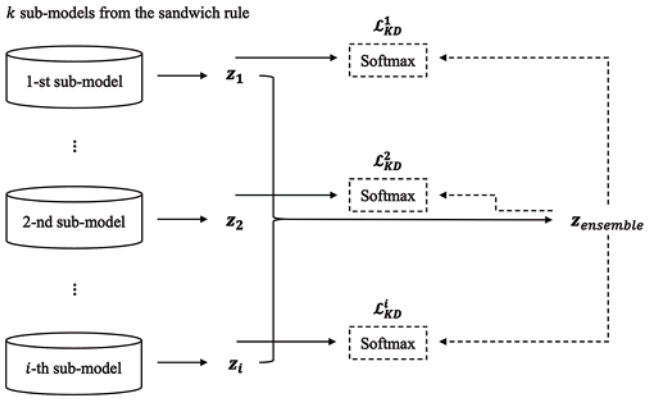


Figure 3: The process of the proposed sandwich rule with an online distillation mechanism. For different sub-models generated by the sandwich rule, respective knowledge distillation (KD) losses are generated. These KD losses are joined with the respective cross-entropy losses

Subsequently, for each sub-model, a knowledge distillation (KD) loss is computed using the Kullback-Leibler divergence (KLD) between the soft targets (**p**) and the soft predictions (**q**). The soft targets are derived from $\mathbf{z}_{ensemble}$, whereas the soft predictions are extracted from one of the sub-models. Formally, this process is articulated as follows:

$$\mathbf{p} = softmax\left(\frac{\mathbf{z}_{ensemble}}{T}\right), \tag{20}$$

$$\mathbf{q} = softmax\left(\frac{\mathbf{z}_i}{T}\right), \tag{21}$$

$$\mathcal{L}_{KD} := T^2 KLD(\mathbf{p}, \mathbf{q}), \tag{22}$$

where T represents the temperature parameter for KLD. The incorporation of a temperature parameter within the softmax function of the KD loss aids in the effective management of the probability distribution. Additionally, to circumvent potential underflow issues, it is advisable to work with soft predictions in the logarithmic space.

Finally, the KD loss is added to the cross-entropy (CE) loss, which is a product of the sandwich rule, as delineated in Eq. (18). Formally, the KD loss, integrated with a trade-off weight denoted as λ, is calculated as follows:

$$\mathcal{L} := \sum_{i=1}^{k} \mathcal{L}_{CE}^{i} + \lambda \mathcal{L}_{KD}^{i}. \tag{23}$$

This integrated approach holds the promise of enhancing the efficacy of on-demand pruning while harnessing the power of distillation, presenting a compelling avenue for optimizing ASR LMs.

4 Experiments

In this section, this study assesses the effectiveness of the proposed joint pruning and online distillation method for the Mogrifier LSTM LM using the LibriSpeech dataset [62], a commonly used dataset in ASR tasks. Sections 4.1 and 4.2 outline the evaluation metrics and datasets employed in this study, respectively. Sections 4.3 and 4.4 provide comprehensive details about the proposed model specifications and the training process. Finally, Section 4.5 presents the experimental results.

4.1 Evaluation Metrics

This study employs two key evaluation metrics: perplexity (PPL) and word error rate (WER). Perplexity is a widely utilized metric in the field of natural language processing, particularly for assessing the performance of LMs. It quantifies how effectively the probability model predicts the sample and is typically defined for discrete probability distributions. The formula for perplexity is as follows:

$$PPL(W) = P(w_1, w_2, \ldots, w_n)^{-\frac{1}{N}}, \tag{24}$$

where W is the test set composed of N words. A lower perplexity score indicates better generalization performance.

WER is a standard metric for assessing the performance of a speech recognition system. It measures the minimum number of operations (substitutions (S), deletions (D), and insertions (I)) required to transform a system output into the reference output. The formula for WER is:

$$WER = \frac{S + D + I}{N}, \tag{25}$$

where N is the number of words in the reference. A lower WER signifies better system performance.

4.2 Dataset Setup

This study utilizes the LibriSpeech ASR and LM datasets to train the English ASR and LM models, as outlined in Table 1. The LibriSpeech dataset is well-known in the ASR field and is publicly

available. It comprises a wide range of audiobooks narrated by various native English speakers, earning acclaim for its high quality and diverse content. In particular, the text-only LM dataset consists of approximately 40 million normalized sentences sourced from 14,500 public domain books.

Table 1: Statistics for the LibriSpeech automatic speech recognition (ASR) and language model (LM) datasets used in this study

Dataset name	Hours	Per-speaker min	Total speakers	Sentences
ASR train				
: train-clean-100	100.6	25	251	
: train-clean-360	363.6	25	921	
: train-other-500	496.7	30	1166	
ASR test				
: dev-clean	5.4	8	40	
: dev-other	5.3	10	33	
: test-clean	5.4	8	40	
: test-other	5.1	10	33	
LM train				40,418,261
LM test				5,558

In this study, both the ASR and text-only LM datasets in this study are tokenized into 1,024 subwords using SentencePiece [63]. To evaluate the PPL of the LMs, approximately 6K sentences are randomly selected from this dataset, while the rest are allocated for LM training. When assessing the WER of the ASR, the study utilizes four partitions of the LibriSpeech dataset. Each subset comprises approximately 3K utterances that exhibit diverse characteristics, including background noise and non-native speakers. These subsets have been deliberately designed to evaluate ASR models in real-world, less controlled scenarios, effectively simulating the challenging conditions commonly encountered in casual conversational speech.

4.3 Model Setup

All LSTM-based LMs utilized in this study have 512 input and hidden dimensions, as specified in Tables 2 and 3. Initially, eight distinct Mogrifier LSTM LMs are trained from the ground up, each featuring a different number of rounds, spanning from one to eight. Furthermore, an eight-round Mogrifier LSTM is trained using on-demand pruning, as described in Section 3.2. Additionally, the proposed eight-round Mogrifier LSTMs are trained using a combined mechanism of the sandwich rule and online distillation, as explained in Section 3.3.

4.4 Training Details

For the training of these LMs, a batch size of 32 is employed, with each training example containing 512 tokens. All models are trained for a maximum of 300K iterations, and the checkpoint with the lowest loss value is selected. It is confirmed that all models converge within 300K steps. The Adam optimizer [64] is utilized, and the learning rate is gradually warmed up from 10^{-7} to a peak of 10^{-3}. Subsequently, it is decayed using a cosine scheduler with a weight decay of 10^{-2}. The temperature

for the distillation loss is set at 2, and the trade-off weight between the KD loss and the CE loss is fixed at 0.5. All models are trained using the Fairseq [65] framework with a single NVIDIA V100 GPU.

Table 2: Structure of vanilla long short-term memory (LSTM) used in this study

Layer name	Input size	Output size	Use of bias
Embedding	1024	512	–
LSTM × 4	512	512	–
Full connected	512	1024	True

Table 3: Structure of Mogrifier long short-term memory (LSTM) used in this study

Layer name	Input size	Output size	Use of bias
Embedding	1024	512	–
Mogrifier LSTM × 2	–	–	–
>Linear × 8	512	512	True
>LSTM	512	512	–
Full connected	512	1024	True

For the training of AMs, this study employs a 16-layer Conformer-M encoder [3] trained with the CTC loss [6]. In all experiments, beam search decoding is utilized in combination with LMs. The decoding algorithm follows a frame-synchronous approach [10], with the beam size set to 5, and the LM weight is fixed at 0.4, a value determined to be optimal through preliminary experiments.

4.5 Experimental Results

This study conducts three distinct sets of experiments to evaluate the performance of the proposed approach. In the first set, a comparison is made between the performance of the vanilla LSTM LM and the Mogrifier LSTM LM in the ASR task. The second set of experiments demonstrates the efficiency of the proposed joint on-demand pruning and online distillation method by comparing it to previous on-demand pruning strategies that serve as a baseline. As the third experiment, an ablation study of the proposed method is conducted.

4.5.1 Comparison of LSTM LMs

In the initial step, this study compares the performance of shallow fusion using the vanilla LSTM LM and the Mogrifier LSTM LM. To ensure a fair comparison, both LSTM LMs have the same model parameters. The vanilla LSTM consists of four layers, while the Mogrifier LSTM comprises two layers with eight rounds each.

As elaborated in Section 3.1, the Mogrifier LSTM LM incorporates a distinct module called rounds, which facilitates mutual gating between the current inputs and previous hidden states. In this study, the rounds of the Mogrifier LSTM LM serve as targets for on-demand pruning. Table 4 illustrates the effectiveness of pruning Mogrifier rounds and provides a comparative analysis with the conventional method of pruning layers from a vanilla LSTM LM. Notably, even without the implementation of specific training techniques for pruning, the Mogrifier LSTM LM slightly

outperforms its vanilla counterpart. When a layer is pruned from the vanilla LSTM LM, performance dramatically declines, occasionally yielding results even worse than ASR performance without any LMs. In contrast, while performance does decrease when rounds are pruned from the Mogrifier LSTM LM, the LM maintains its efficiency.

Table 4: Comparison of shallow fusion with language models (LMs) which comprise vanilla and Mogrifier long short-term memory (LSTM) layers respectively in the **test-other** partition of LibriSpeech

Type	Training layer/round	Inference layer/round	#param	PPL	WER (%)
Greedy (ASR w/o LM)				N/A	7.53
Vanilla LSTM	4/0	4/0	9.5M	44.38	6.71
	4/0	3/0	7.4M	1089.04	10.66
Mogrifier LSTM	2/8	2/8	9.5M	41.77	**6.62**
	2/8	2/4	7.4M	263.0	**7.34**

Furthermore, a comparative analysis has been conducted, comparing the Mogrifier LSTM LM against a variety of alternative architectures, as presented in Table 5. This comprehensive evaluation effectively highlights the capabilities of the Mogrifier LSTM LM in the context of the ASR task. The results not only showcase its ability to compete with established architectures but also position it as a viable and promising choice for enhancing ASR performance.

Table 5: Comparison of shallow fusion with language models (LMs) which comprise several architectures and Mogrifier long short-term memory (LSTM) LM respectively in the **test-other** partition of LibriSpeech

Type	Layer/dimension/head/round	#param	WER (%)
Greedy (ASR w/o LM)			7.53
GPT-2 [53]	4/512/8/0	13.4M	7.32
TFM-XL [22]	4/512/8/0	12.9M	6.71
TFM-XL [22]	3/512/8/0	9.8M	6.96
Mogrifier LSTM	2/512/0/8	9.5M	**6.62**

4.5.2 Comparison of On-Demand Pruning Strategies

The following series of experiments demonstrate the effectiveness of the suggested joint on-demand pruning and online distillation method by comparing it to previous on-demand pruning strategies, as shown in Tables 6–9. As elaborated in Section 4.3, these experiments employ the Mogrifier LSTM LM with two layers and eight rounds.

Table 6: Comparison of shallow fusion with Mogrifier long short-term memory language models (LMs) with on-demand pruning strategies and the proposed joint online distillation on the **dev-clean** partition of LibriSpeech under various run-time scenarios. The parameter $\boldsymbol{k}$ in the sandwich rule represents the number of losses computed per training iteration. Here, this study determines that setting $\boldsymbol{k} = \mathbf{2}$ or $\boldsymbol{k} = \mathbf{3}$ is adequate to achieve comparable accuracy to the baseline

Inference #round	1	2	3	4	5	6	7	8
#param of LMs (M)	5.8	6.3	6.8	7.4	7.9	8.4	8.9	9.5
Individuals	2.60	2.57	2.53	2.54	2.54	2.55	2.53	2.52
Random	2.62	2.61	2.59	2.59	2.58	2.58	2.58	2.58
Sandwich ($k = 3$)	2.61	2.60	2.59	2.60	2.59	2.58	2.57	2.57
+ Joint distillation (proposed)	**2.58**	**2.56**	**2.55**	**2.56**	**2.56**	**2.55**	**2.56**	**2.56**
Sandwich ($k = 2$)	2.59	2.58	2.55	2.57	2.58	2.56	2.57	2.57
+ Joint distillation (proposed)	**2.58**	**2.56**	**2.56**	**2.56**	**2.56**	**2.56**	**2.54**	**2.53**

Table 7: Comparison of shallow fusion with Mogrifier long short-term memory language models (LMs) with on-demand pruning strategies and the proposed joint online distillation on the **dev-other** partition of LibriSpeech under various run-time scenarios. The parameter $\boldsymbol{k}$ in the sandwich rule represents the number of losses computed per training iteration. Here, this study determines that setting $\boldsymbol{k} = \mathbf{2}$ or $\boldsymbol{k} = \mathbf{3}$ is adequate to achieve comparable accuracy to the baseline

Inference #round	1	2	3	4	5	6	7	8
#param of LMs (M)	5.8	6.3	6.8	7.4	7.9	8.4	8.9	9.5
Individuals	6.58	6.56	6.57	6.52	6.56	6.56	6.50	6.58
Random	6.70	6.61	6.59	6.58	6.57	6.55	6.55	6.58
Sandwich ($k = 3$)	6.65	6.63	6.64	6.62	6.60	6.60	6.57	6.57
+ Joint distillation (proposed)	**6.62**	**6.59**	**6.58**	**6.55**	**6.55**	**6.53**	**6.53**	**6.53**
Sandwich ($k = 2$)	6.61	6.62	6.60	6.58	6.62	6.61	6.59	6.57
+ Joint distillation (proposed)	**6.57**	**6.57**	**6.57**	**6.56**	**6.55**	**6.53**	**6.52**	**6.54**

In the second row of these tables, individual Mogrifier LSTMs are presented, with each one being independently trained for rounds 1 through 8, without employing on-demand pruning. The number of parameters for these models varies from 5.8 million to 9.5 million, and during inference, it remains fixed to match the round used during training. This approach operates under the assumption that there are adequate resources available to train each model individually. While this approach may yield reliable performance, it is not practical due to its inefficiency.

Starting from the third row, this study introduces baseline models trained using on-demand pruning strategies. For the random method, this study randomly selects one of the eight rounds for each iteration during training. The overall performance of the random method is less consistent compared to the individual models, especially at the lower and upper bounds (with rounds of 1 or 8). Conversely, when employing the sandwich rule, this study observes a more stable WER at lower rounds compared to the random method. This stability is particularly evident at the lower and upper bounds, where

the performance gap relative to individual models is significantly reduced compared to the random method. This study uses $k = 3$ and $k = 2$ for the sandwich rule, where $k = 3$ includes cases that are randomly selected at each iteration during training, as explained in Section 3.2. It is important to note that finding the right balance between the value of k and accuracy is crucial, as increasing k results in higher computational costs.

Table 8: Comparison of shallow fusion with Mogrifier long short-term memory language models (LMs) with on-demand pruning strategies and the proposed joint online distillation on the **test-clean** partition of LibriSpeech under various run-time scenarios. The parameter $\boldsymbol{k}$ in the sandwich rule represents the number of losses computed per training iteration. Here, this study determines that setting $\boldsymbol{k} = \mathbf{2}$ or $\boldsymbol{k} = \mathbf{3}$ is adequate to achieve comparable accuracy to the baseline

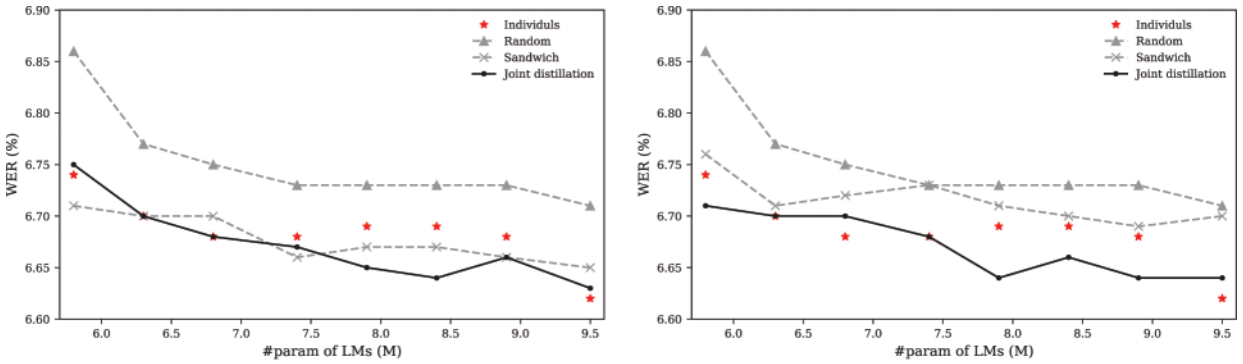
Inference #round	1	2	3	4	5	6	7	8
#param of LMs (M)	5.8	6.3	6.8	7.4	7.9	8.4	8.9	9.5
Individuals	2.79	2.76	2.78	2.79	2.74	2.73	2.78	2.73
Random	2.83	2.79	2.79	2.78	2.77	2.77	2.76	2.77
Sandwich ($k = 3$)	2.81	2.80	2.79	2.78	2.79	2.76	2.76	2.76
+ Joint distillation (proposed)	**2.81**	**2.78**	**2.78**	**2.78**	**2.76**	**2.76**	**2.75**	**2.77**
Sandwich ($k = 2$)	2.78	2.77	2.77	2.78	2.78	2.76	2.75	2.75
+ Joint distillation (proposed)	**2.78**	**2.74**	**2.74**	**2.74**	**2.73**	**2.74**	**2.76**	**2.74**

Table 9: Comparison of shallow fusion with Mogrifier long short-term memory language models (LMs) with on-demand pruning strategies and the proposed joint online distillation on the **test-other** partition of LibriSpeech under various run-time scenarios. The parameter $\boldsymbol{k}$ in the sandwich rule represents the number of losses computed per training iteration. Here, this study determines that setting $\boldsymbol{k} = \mathbf{2}$ or $\boldsymbol{k} = \mathbf{3}$ is adequate to achieve comparable accuracy to the baseline

Inference #round	1	2	3	4	5	6	7	8
#param of LMs (M)	5.8	6.3	6.8	7.4	7.9	8.4	8.9	9.5
Individuals	6.74	6.70	6.68	6.68	6.69	6.69	6.68	6.62
Random	6.86	6.77	6.75	6.73	6.73	6.73	6.73	6.71
Sandwich ($k = 3$)	6.71	6.70	6.70	6.66	6.67	6.67	6.66	6.65
+ Joint distillation (proposed)	**6.75**	**6.70**	**6.68**	**6.67**	**6.65**	**6.64**	**6.66**	**6.63**
Sandwich ($k = 2$)	6.76	6.71	6.72	6.73	6.71	6.70	6.69	6.70
+ Joint distillation (proposed)	**6.71**	**6.70**	**6.70**	**6.68**	**6.64**	**6.66**	**6.64**	**6.64**

Nevertheless, there are trade-offs to consider, especially regarding the potential for unstable accuracy when using sandwich rule-based pruning strategies to achieve a smaller model size. To address this concern, this study incorporates joint online distillation alongside the sandwich rule, as illustrated in these tables and Fig. 4 (Fig. 4 is included to provide an intuitive comparison between the sandwich rule and joint online distillation methods, using the data from Table 9). The experimental results of the proposed joint online distillation indicate an overall improvement in performance compared to using the sandwich rule alone. The proposed setup not only reduces the number of parameters

by approximately 39% but also boosts performance, outperforming the models trained individually. Moreover, this approach showcases its robustness, sustaining its performance even when pruned down to models with fewer parameters.



(a) $k = 3$ for the sandwich rule **(b)** $k = 2$ for the sandwich rule

Figure 4: A comparison between the sandwich rule and the proposed joint online distillation methods on the **test-other** partition of the LibriSpeech under varying language model (LM) parameters

Upon a detailed analysis of these experimental results, several insights emerge that confirm the effectiveness of integrating the online distillation mechanism. The data clearly shows that knowledge transfer, facilitated by online distillation, takes place seamlessly during the training phase. This manifests in the smaller model's enhanced ability to absorb and reflect valuable information from its larger counterpart, leading to improved model accuracy and stability.

Moreover, the inherent flexibility of the model, achieved through on-demand construction, is evident in the experiments. The model showcases its adaptability, demonstrating its capability to dynamically adjust based on situational needs. This flexibility ensures that the model remains versatile, and capable of catering to different devices or scenarios without requiring separate training for each context.

Additionally, this study analyzes several samples of ASR recognition results from both the sandwich rule method (with $k = 3$) and the proposed joint online distillation method, as presented in Table 10. This study focuses on cases where the online distillation mechanism has been jointly applied and observes that, in certain instances, this approach addresses issues such as deletion, insertion, and replacement, leading to outcomes that align with the correct answers.

Table 10: Qualitative analysis of automatic speech recognition results for the sandwich rule and the proposed joint online distillation method

Answer	Sandwich	Sandwich + **Joint distillation (proposed)**
ho this will bang it soundly	oh this was **banging** soundly	oh this was **bang** it soundly
well cacked well sung	well cackle well **so**	well cackle well **sung**
a man wishing to go to a certain place comes to where the road divides	a man wishing to go to a certain place come **toward** the road divides	a man wishing to go to a certain place come **to where** the road divides

4.5.3 Ablation Study of the Joint On-Demand Pruning and Online Distillation

This study conducts an ablation study on the proposed method, which combines on-demand pruning and online distillation. As explained in Section 3.3, this study utilizes the ensemble logit of all sub-models to obtain soft targets in joint online distillation. To assess the effectiveness of this collaborative learning strategy, this study replaces the ensemble logit with the logit from the model that exhibits the smallest loss when obtaining the soft target.

The experimental results demonstrate that employing ensembles for joint online distillation generally yields comparable or superior performance to using the model with the lowest loss, as indicated in Tables 11–14 and Fig. 5 (Fig. 5 offers an intuitive comparison between the ensemble logit and the minimum logit when applying joint online distillation methods, utilizing the data from Table 14). This implies that the logits generated by the sub-models constructed based on the k value of the sandwich rule closely influence one another.

Table 11: Comparison of ensemble and minimum when applying proposed joint online distillation methods on the **dev-clean** partition of LibriSpeech under various run-time scenarios. The parameter ***k*** in the sandwich rule represents the number of losses computed per training iteration

Inference #round	1	2	3	4	5	6	7	8
#param of LMs (M)	5.8	6.3	6.8	7.4	7.9	8.4	8.9	9.5
Sandwich ($k = 3$)	2.61	2.60	2.59	2.60	2.59	2.58	2.57	2.57
+ Joint distillation (minimum)	2.62	2.59	2.59	2.59	2.57	2.57	2.56	2.56
+ Joint distillation (ensemble)	**2.58**	**2.56**	**2.55**	**2.56**	**2.56**	**2.55**	**2.56**	**2.56**

Table 12: Comparison of ensemble and minimum when applying proposed joint online distillation methods on the **dev-other** partition of LibriSpeech under various run-time scenarios. The parameter ***k*** in the sandwich rule represents the number of losses computed per training iteration

Inference #round	1	2	3	4	5	6	7	8
#param of LMs (M)	5.8	6.3	6.8	7.4	7.9	8.4	8.9	9.5
Sandwich ($k = 3$)	6.65	6.63	6.64	6.62	6.60	6.60	6.57	6.57
+ Joint distillation (minimum)	6.61	6.58	6.57	6.52	6.52	6.50	6.52	6.51
+ Joint distillation (ensemble)	**6.62**	**6.59**	**6.58**	**6.55**	**6.55**	**6.53**	**6.53**	**6.53**

In summary, the fusion of on-demand pruning strategies with the online distillation mechanism empowers us to dynamically adjust the number of gates in the Mogrifier LSTM at runtime while maintaining high levels of accuracy. This adaptability streamlines the deployment of ASR model across a diverse array of devices, spanning from high-end to entry-level, all without the need for training multiple models of varying sizes tailored to each specific scenario. The effectiveness of our proposed method shines through in our evaluations on the LibriSpeech dataset. Nevertheless, it is imperative to undertake further validation across a spectrum of ASR and LM models. This crucial step will serve as the focal point of our forthcoming research endeavors, ensuring the robustness and versatility of our approach in diverse linguistic and acoustic environments.

Table 13: Comparison of ensemble and minimum when applying proposed joint online distillation methods on the **test-clean** partition of LibriSpeech under various run-time scenarios. The parameter ***k*** in the sandwich rule represents the number of losses computed per training iteration

Inference #round	1	2	3	4	5	6	7	8
#param of LMs (M)	5.8	6.3	6.8	7.4	7.9	8.4	8.9	9.5
Sandwich ($k = 3$)	2.81	2.80	2.79	2.78	2.79	2.76	2.76	2.76
+ Joint distillation (minimum)	2.82	2.80	2.78	2.78	2.75	2.75	2.76	2.76
+ Joint distillation (ensemble)	**2.81**	**2.78**	**2.78**	**2.78**	**2.76**	**2.76**	**2.75**	**2.77**

Table 14: Comparison of ensemble and minimum when applying proposed joint online distillation methods on the **test-other** partition of LibriSpeech under various run-time scenarios. The parameter ***k*** in the sandwich rule represents the number of losses computed per training iteration

Inference #round	1	2	3	4	5	6	7	8
#param of LMs (M)	5.8	6.3	6.8	7.4	7.9	8.4	8.9	9.5
Individuals	2.60	2.57	2.53	2.54	2.54	2.55	2.53	2.52
Sandwich ($k = 3$)	6.71	6.70	6.70	6.66	6.67	6.67	6.66	6.65
+ Joint distillation (minimum)	6.76	6.74	6.72	6.70	6.69	6.68	6.69	6.68
+ Joint distillation (ensemble)	**6.75**	**6.70**	**6.68**	**6.67**	**6.65**	**6.64**	**6.66**	**6.63**

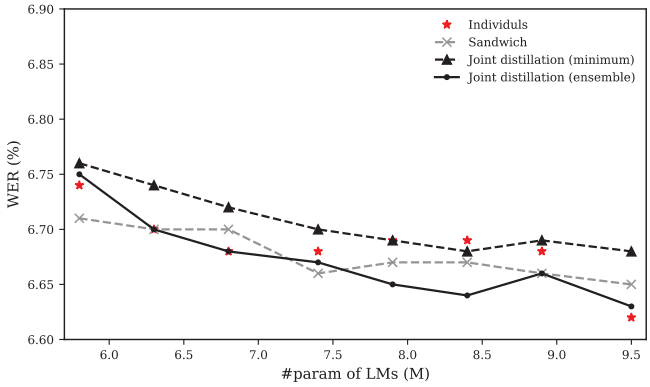


Figure 5: An ablation study for the proposed joint online distillation using an ensemble mechanism on the **test-other** partition of the LibriSpeech under varying language model (LM) parameters. Here, $k = 3$ for the sandwich rule

5 Conclusions

The approach proposed in this study offers an efficient and robust solution to the challenge of deploying ASR models across various conditions, encompassing a wide spectrum of computational resources. By utilizing the Mogrifier LSTM LM in conjunction with the proposed joint on-demand pruning and online distillation, this study effectively adjusts the number of gates in the model while maintaining high accuracy. The experimental results demonstrate that the proposed approach surpasses the performance of the vanilla LSTM LM and maintains stable performance compared to

the sandwich rule-based on-demand pruning strategies. Moreover, the optimal configuration proposed in this study reduces the number of parameters by approximately 39%, facilitating efficient deployment on both high-end and entry-level devices.

Acknowledgement: The authors would like to express their gratitude to the editors and reviewers for their thorough review and valuable recommendations.

Funding Statement: This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-00377, Development of Intelligent Analysis and Classification Based Contents Class Categorization Technique to Prevent Imprudent Harmful Media Distribution).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: S. Seo, J-H. Kim; data collection: S. Seo; analysis and interpretation of results: S. Seo, J-H. Kim; draft manuscript preparation: S. Seo. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: All the data used in this study are publicly available, and readers can access them freely.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] A. Baevski, Y. Zhou, A. Mohamed and M. Auli, "Wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. of NeurIPS*, Virtual Event, pp. 12449–12460, 2020.

[2] Q. Zhang, H. Lu, H. Sak, A. Tripathi, E. McDermott *et al.,* "Transformer transducer: A streamable speech recognition model with transformer encoders and RNN-T loss," in *Proc. of ICASSP*, Barcelona, Spain, pp. 7829–7833, 2020.

[3] A. Gulati, J. Qin, C. C. Chiu, N. Parmar, Y. Zhang *et al.,* "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. of INTERSPEECH*, Shanghai, China, pp. 5036–5040, 2020.

[4] E. Tsunoo, Y. Kashiwagi, C. Narisetty and S. Watanabe, "Residual language model for end-to-end speech recognition," in *Proc. of INTERSPEECH*, Incheon, South Korea, pp. 3899–3903, 2022.

[5] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu *et al.,* "WavLM: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[6] A. Graves, S. Fernández, F. Gomez and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. of ICML*, Pittsburgh, PA, USA, pp. 369–376, 2006.

[7] A. Graves, "Sequence transduction with recurrent neural networks," in *Proc. of ICML Workshop on Representation Learning*, Edinburgh, Scotland, 2012.

[8] W. Chan, N. Jaitly, Q. Le and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. of ICASSP*, Shanghai, China, pp. 4960–4964, 2016.

[9] L. Dong, S. Xu and B. Xu, "Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition," in *Proc. of ICASSP*, Calgary, AB, Canada, pp. 5884–5888, 2018.

[10] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," *Proceedings of Machine Learning Research*, vol. 32, no. 2, pp. 1764–1772, 2014.

[11] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos *et al.,* "Deep speech: Scaling up end-to-end speech recognition," arXiv:1412.5567, 2014.

[12] D. Zhao, T. N. Sainath, D. Rybach, P. Rondon, D. Bhatia *et al.,* "Shallow-fusion end-to-end contextual biasing," in *Proc. of INTERSPEECH*, Graz, Austria, pp. 1418–1422, 2019.
[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[14] P. Suthanthira and S. Karthika, "RDNN: Rumor detection neural network for veracity analysis in social media text," *KSII Transactions on Internet and Information Systems*, vol. 16, no. 12, pp. 3868–3888, 2022.
[15] B. Vijayalakshmi, S. T. Ramya and K. Ramar, "Multivariate congestion prediction using stacked LSTM autoencoder based bidirectional LSTM model," *KSII Transactions on Internet and Information Systems*, vol. 17, no. 1, pp. 216–238, 2023.
[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones *et al.,* "Attention is all you need," in *Proc. of NIPS*, Long Beach, CA, USA, pp. 5998–6008, 2017.
[17] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le *et al.,* "Transformer-XL: Attentive language models beyond a fixed-length context," in *Proc. of ACL*, Florence, Italy, pp. 2928–2988, 2019.
[18] J. Lee, N. Chirkov, E. Ignasheva, Y. Pisarchyk, M. Shieh *et al.,* "On-device neural net inference with mobile GPUs," in *Proc. of CVPR Workshop on Efficient Deep Learning for Computer Vision*, Long Beach, CA, USA, 2019.
[19] A. Ignatov, R. Timofte, A. Kulik, S. Yang, K. Wang *et al.,* "AI benchmark: All about deep learning on smartphones in 2019," in *Proc. of ICCV Workshop*, Seoul, South Korea, pp. 3617–3635, 2019.
[20] J. Park, S. Jin, J. Park, S. Kim, D. Sandhyana *et al.,* "Conformer-based on-device streaming speech recognition with KD compression and two-pass architecture," in *Proc. of SLT*, Doha, Qatar, pp. 92–99, 2023.
[21] J. Yu, L. Yang, N. Xu, J. Yang and T. Huang, "Slimmable neural networks," in *Proc. of ICLR*, New Orleans, LA, USA, 2019.
[22] J. Yu and T. S. Huang, "Universally slimmable networks and improved training techniques," in *Proc. of ICCV*, Seoul, South Korea, pp. 1803–1811, 2019.
[23] C. Li, G. Wang, B. Wang, X. Liang, Z. Li *et al.,* "Dynamic slimmable network," in *Proc. of CVPR*, Virtual Event, pp. 8607–8617, 2021.
[24] G. Huang, Y. Sun, Z. Liu, D. Sedra and K. Q. Weinberger, "Deep networks with stochastic depth," in *Proc. of ECCV*, Amsterdam, The Netherlands, pp. 646–661, 2016.
[25] A. Fan, E. Grave and A. Joulin, "Reducing transformer depth on demand with structured dropout," in *Proc. of ICLR*, Virtual Event, 2020.
[26] J. Lee, J. Kang and S. Watanabe, "Layer pruning on demand with intermediate CTC," in *Proc. of INTERSPEECH*, Brno, Czechia, pp. 3745–3749, 2021.
[27] A. Vyas, W. N. Hsu, M. Auli and A. Baevski, "On-demand compute reduction with stochastic wav2vec 2.0," in *Proc. of INTERSPEECH*, Incheon, South Korea, pp. 3048–3052, 2022.
[28] S. Han, J. Pool, J. Tran and W. J. Dally, "Learning both weights and connections for efficient neural network," in *Proc. of NIPS*, Montreal, QC, Canada, pp. 1135–1143, 2015.
[29] Z. Wu, T. Nagarajan, A. Kumar, S. Rennie, L. S. Davis *et al.,* "Blockdrop: Dynamic inference paths in residual networks," in *Proc. of CVPR*, Salt Lake City, UT, USA, pp. 8817–8826, 2018.
[30] Z. Liu, M. Sun, T. Zhou, G. Huang and T. Darrell, "Rethinking the value of network pruning," in *Proc. of ICLR*, New Orleans, LA, USA, 2019.
[31] Z. Chen, Z. Xie, Z. Wang, T. Xu and Z. Zhang, "Filter contribution recycle: Boosting model pruning with small norm filters," *KSII Transactions on Internet and Information Systems*, vol. 16, no. 11, pp. 3507–3522, 2022.
[32] Y. Mao, B. Song, Z. Zhang, W. Yang and Y. Lan, "Multi-classification sensitive image detection method based on lightweight convolutional neural network," *KSII Transactions on Internet and Information Systems*, vol. 17, no. 5, pp. 1433–1449, 2023.
[33] G. Melis, "Circling back to recurrent models of language," arXiv:2211.01848, 2022.
[34] G. Melis, T. Kocisk and P. Blunsom, "Mogrifier LSTM," in *Proc. of ICLR*, New Orleans, LA, USA, 2019.

[35] M. Suzuki, N. Itoh, T. Nagano, G. Kurata and S. Thomas, "Improvements to n-gram language model using text generated from neural language model," in *Proc. of ICASSP*, Brighton, UK, pp. 7245–7249, 2019.
[36] Y. Bengio, R. Ducharme, P. Vincent and C. Jauvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.
[37] G. Zhou, T. He, J. Zhao and P. Hu, "Learning continuous word embedding with metadata for question retrieval in community question answering," in *Proc. of ACL*, Beijing, China, pp. 250–259, 2015.
[38] T. Mikolov, M. Karafiat, L. Burget, J. Cernocky and S. Khudanpur, "Recurrent neural network based language model," in *Proc. of INTERSPEECH*, Makuhari, Japan, pp. 1045–1048, 2010.
[39] X. Chen, A. Ragni, X. Liu and M. Gales, "Investigating bidirectional recurrent neural network language models for speech recognition," in *Proc. of INTERSPEECH*, Stockholm, Sweden, pp. 269–273, 2017.
[40] R. Pascanu, T. Mikolov and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. ICML*, Atlanta, GA, USA, pp. 1310–1318, 2013.
[41] E. Arisoy, A. Sethy, B. Ramabhadran and S. Chen, "Bidirectional recurrent neural network language models for automatic speech recognition," in *Proc. of ICASSP*, South Brisbane, QLD, Australia, pp. 5421–5425, 2015.
[42] Y. Zhang, X. Wang and H. Tang, "An improved Elman neural network with piecewise weighted gradient for time series prediction," *Neurocomputing*, vol. 359, no. 2, pp. 199–208, 2019.
[43] U. Khandelwal, H. He, P. Qi and D. Jurafsky, "Sharp nearby, fuzzy far away: How neural language models use context," in *Proc. of ACL*, Melbourne, VIC, Australia, pp. 284–294, 2018.
[44] K. Xu, J. L. Ba, R. Kiros, K. Cho, A. Courville *et al.,* "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. of ICML*, Lile, France, pp. 2048–2057, 2015.
[45] R. Al-Rfou, D. Choe, N. Constant, M. Guo and L. Jones, "Character-level language modeling with deeper self-attention," in *Proc. of AAAI*, Honolulu, HI, USA, pp. 3159–3166, 2019.
[46] S. Duan, H. Zhao, J. Zhou and R. Wang, "Syntax-aware transformer encoder for neural machine translation," in *Proc. of IALP*, Shanghai, China, pp. 396–401, 2019.
[47] J. Devlin, M. W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. of NAACL*, Minneapolis, MN, USA, pp. 4171–4186, 2019.
[48] Y. Qu, P. Liu, W. Song, L. Liu and M. Cheng, "A text generation and prediction system: Pre-training on new corpora using BERT and GPT-2," in *Proc. of ICEIEC*, Beijing, China, pp. 323–326, 2020.
[49] G. Hinton, O. Vinyals and J. Dean, "Distilling the knowledge in a neural network," in *Proc. of NIPS*, Montreal, QC, Canada, 2014.
[50] S. Sun, Y. Cheng, Z. Gan and J. Liu, "Patient knowledge distillation for bert model compression," in *Proc. of EMNLP-IJCNLP*, Hong Kong, China, pp. 4314–4323, 2019.
[51] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen *et al.,* "TinyBERT: Distilling bert for natural language understanding," in *Proc. of EMNLP*, Virtual Event, pp. 4163–4174, 2020.
[52] J. Wang, Y. Wu, S. He, P. K. Sharma, X. Yu *et al.,* "Lightweight single image super-resolution convolution neural network in portable device," *KSII Transactions on Internet and Information Systems*, vol. 15, no. 11, pp. 4065–4083, 2021.
[53] X. Lan, X. Zhu and S. Gong, "Knowledge distillation by on-the-fly native ensemble," in *Proc. of NeurIPS*, Montreal, QC, Canada, pp. 7528–7538, 2018.
[54] Y. Zhang, T. Xiang, T. M. Hospedales and H. Lu, "Deep mutual learning," in *Proc. of CVPR*, Salt Lake City, UT, USA, pp. 4320–4328, 2018.
[55] Q. Guo, X. Wang, Y. Wu, Z. Yu, D. Liang *et al.,* "Online knowledge distillation via collaborative learning," in *Proc. of CVPR*, Seattle, WA, USA, pp. 11017–11026, 2020.
[56] B. Cui, Y. Li and Z. Zhang, "Joint structured pruning and dense knowledge distillation for efficient transformer model compression," *Neurocomputing*, vol. 458, pp. 56–69, 2021.
[57] R. Wang, S. Wan, W. Zhang, C. Zhang, Y. Li *et al.,* "Progressive multi-level distillation learning for pruning network," *Complex & Intelligent Systems*, vol. 9, pp. 5779–5791, 2023.
[58] V. S. Lodagala, S. Ghosh and S. Umesh, "PADA: Pruning assisted domain adaptation for self-supervised speech representations," in *Proc. of SLT*, Doha, Qatar, pp. 136–143, 2023.

[59] Y. Peng, K. Kim, F. Wu, P. Sridhar and S. Wantanabe, "Structured pruning of self-supervised pre-trained models for speech recognition and understanding," in *Proc. of ICASSP*, Rhodes Island, Greece, pp. 1–5, 2023.
[60] Y. Peng, Y. Sudo, S. Muhammad and S. Watanabe, "DPHuBERT: Joint distillation and pruning of self-supervised speech models," in *Proc. of INTERSPEECH*, Dublin, Ireland, pp. 62–66, 2023.
[61] S. Merity, "Single headed attention RNN: Stop thinking with your head," arXiv:1911.11423, 2019.
[62] V. Panayotov, G. Chen, D. Povey and S. Khudanpur, "LibriSpeech: An ASR corpus based on public domain audio books," in *Proc. of ICASSP*, Brisbane, QLD, Australia, pp. 5206–5210, 2015.
[63] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *Proc. of EMNLP (Demonstration)*, Brussels, Belgium, 2018.
[64] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980, 2014.
[65] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross *et al.,* "Fairseq: A fast, extensible toolkit for sequence modeling," arXiv:1904.01038, 2019.