BiCycle: Group-wise Recursive Transformer Based on ASR Mechanism

Min Ho Jang, Eun Seo Seo, Jin Young Kim, Hyeongsoo Lim, Ji Won Yoon*

Department of AI, Chung-Ang University, Seoul, South Korea
{sunbi8534, jeo0534, wlsdud338, andrew1001, jiwonyoon}@cau.ac.kr

Abstract

Recursive transformer (RT) is a promising parameter-sharing technique for reducing computational burden of large-scale model. While RT has been successfully applied to large language models (LLMs), its effectiveness in automatic speech recognition (ASR) remains limited, despite the parallel trend of model scaling in the speech domain. In this paper, we reveal that conventional RT designs for LLMs are suboptimal for speech recognition, primarily because they do not fully consider the layer-wise specialization inherent in the ASR architecture, where lower layers focus on phonetic features and upper layers capture linguistic localization. To address this, we propose BiCycle, a novel RT scheme tailored for ASR. In particular, we firstly analyze attention patterns in a pre-trained ASR model to divide its layers into phonetic and linguistic groups. BiCycle then constructs an efficient RT model by transferring the pre-trained model's weights in a step-wise manner and applies recursion separately to the phonetic and linguistic groups, preventing conflicts between their roles. To further maximize BiCycle's performance, we propose group-wise feature distillation (GFD), which performs feature-level knowledge distillation (KD) between the teacher and student models' phonetic and linguistic groups, thereby effectively transferring the teacher model's knowledge tailored to each group's role. Extensive experimental results confirm that the proposed method not only preserves the original ASR mechanism but also outperforms conventional RT approaches.

Introduction

In recent years, under the paradigm of scaling laws (Kaplan et al. 2020), model sizes have grown exponentially to further push performance limits across various domains (Gu et al. 2023; Zhai et al. 2022; Bahri et al. 2021). Although such large-scale models have achieved remarkable breakthroughs (Yang et al. 2025; Liu et al. 2024; Touvron et al. 2023), they are memory- and computation-intensive, posing a critical bottleneck in resource-constrained environments. Parameter sharing offers a promising direction for mitigating this computational burden by reducing redundancy in model parameters (Nouriborji et al. 2023; Shen, Liu, and Xing 2022; Lan et al. 2020). In particular, sharing weights across layers reduces the overall memory footprint, which can lower hardware requirements and enable the use of larger batch sizes, ultimately improving training and inference efficiency (Wang et al. 2025; Cao, Yang, and Zhao 2024).

Among parameter sharing strategies, a widely used technique is the recursive transformer (RT), which repeatedly reuses one or more shared transformer blocks across layers, effectively increasing the model's logical depth. RT has been actively explored in recent research, particularly for large language models (LLMs) (Bae et al. 2025; Li et al. 2025a; Takase and Kiyono 2023), where transformer layers serve as the dominant architectural building blocks and account for the majority of the model's parameters (Minaee et al. 2024; Vaswani et al. 2017). In this context, prior studies have proposed various strategies for initializing RT models using weights from pre-trained, non-shared LLMs, demonstrating successful results in LLMs. These approaches aim to effectively transfer the rich knowledge of the pre-trained LLM to the RT's parameter-sharing architecture, thereby enabling stable training and faster convergence of the model despite the inherent constraints of weight sharing. While RT has shown promise in reducing the size and computational demands of LLMs, its applicability to automatic speech recognition (ASR) remains limited. Given the parallel trend of aggressive model scaling in the speech domain (Chen et al. 2025; Kashiwagi et al. 2025; Song et al. 2024; Zhang et al. 2024), enabling the effective adoption of RT architectures while transferring useful knowledge from pre-trained models is desirable for building more efficient ASR systems.

In this paper, we reveal that existing RT approaches, originally developed for LLMs, are suboptimal for speech recognition, as they do not fully account for the layer-wise specialization inherent in the conventional ASR architecture. Given the distinct roles of different layers, where lower layers focus on phonetic features while upper layers capture higher-level linguistic information (Shim, Choi, and Sung 2022; Yang, Liu, and Lee 2020), naively applying RT design from LLM may not generalize well to the ASR task. Based on this understanding, we propose a novel RT scheme for ASR, namely *BiCycle*. Specifically, BiCycle encapsulates three key components. First, we analyze the attention patterns of the pre-trained ASR model to clearly divide the entire layers into phonetic and linguistic groups. Second, we initialize the RT architecture with the pre-trained parameters in a step-wise fashion. Third, to preserve the ASR

*Corresponding author.

mechanism where phonetic processing precedes linguistic processing, we employ group-wise independent recursion. This approach prevents conflicts between the phonetic and linguistic processing within the recursive structure. Furthermore, we experimentally confirm that BiCycle is highly effective when used with knowledge distillation (KD) (Hinton, Vinyals, and Dean 2014). To further maximize the performance of BiCycle, we propose group-wise feature distillation (GFD). Unlike conventional feature distillation approach for ASR that typically uses only the model's final layer (Yoon et al. 2021; Romero et al. 2015), GFD performs feature distillation between the teacher and student models' phonetic and linguistic groups, thereby effectively transferring the teacher model's knowledge tailored to each group's role.

Extensive experiments are conducted on the LibriSpeech (Panayotov et al. 2015) and Common Voice 7.0 (Ardila et al. 2020) French datasets to demonstrate the efficacy of our proposed method using the connectionist temporal classification (CTC) (Graves et al. 2006) ASR model. Compared to existing RT structures that overlook the speech recognition mechanism, the proposed RT significantly improves the model's performance. In a detailed case study, we show that the proposed RT processes speech in the same manner as a standard ASR model, despite its recursive structure. In addition, GFD achieves superior performance compared to existing feature distillation method by receiving the teacher model's knowledge tailored to the role of each group.

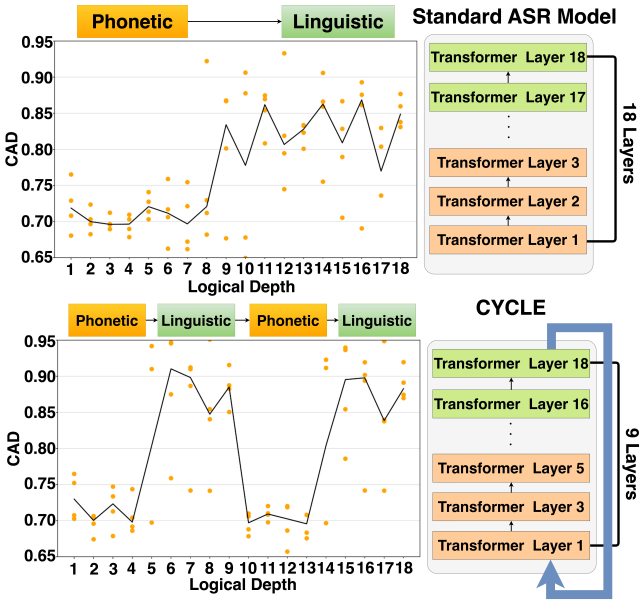


Figure 1: Cumulative attention diagonality (CAD), illustrating the diagonal concentration of attention heads per layer across different model architectures. The top figure shows a standard transformer-based ASR model without RT, while the bottom figure illustrates an ASR model using the CYCLE structure.

Related Work

In LLMs, where transformer layers constitute the vast majority of the model's parameters, RT has emerged as a particularly active area of research in parameter sharing. This approach repeatedly reuses existing transformer layers to effectively expand the model's logical depth without incurring additional parameter cost. Representative RT architectures include the CYCLE structure and SEQUENCE structure proposed by Takase and Kiyono (2023). Specifically, the CYCLE structure treats a block of multiple transformer layers as a single unit, recursively applying computations through this entire block. In contrast, the SEQUENCE structure focuses on reusing a single transformer layer repeatedly, effectively stacking the same layer multiple times to increase depth.

Recently, RT frameworks, particularly those based on the CYCLE structure, have been actively researched. For example, Bae et al. (2025) introduces Relaxed Recursive Transformer, which applies different LoRA (Hu et al. 2022) weights for each recurrence cycle, providing flexibility. Similarly, Li et al. (2025a) proposes the recursive structure that excludes the first and last layers from the recurrent structure, considering their unique roles. This approach also uses zero tokens and a gating network to mitigate unnecessary computations caused by recursion. These previous studies in LLMs successfully apply RT by combining initialization with weights from a pre-trained model and additional components that add flexibility to RT. Meanwhile, in the ASR field, Li et al. (2025b) proposes a Foldable Network based on the SEQUENCE structure that applies a self-distillation process to flexibly adjust the model's logical depth.

While the LLM field has seen significant advancements in RT research by leveraging the weights of pre-trained models, ASR's progress has been different. The Foldable Network, for example, demonstrates a successful use of weights from self-supervised learning (SSL) models (Hsu et al. 2021; Baevski et al. 2020), but there are no widely reported cases of directly using the weights from a general pre-trained ASR model. Consequently, research into RT architectures that leverages pre-trained ASR model weights remains a relatively unexplored area.

Proposed Method

Motivation

In ASR, cumulative attention diagonality (CAD) serves as a tool for analyzing attention patterns by quantifying how much attention heads focus on diagonal elements within the attention matrix, thereby indicating a localized processing behavior (Shim, Choi, and Sung 2022). Examining CAD in the standard ASR model reveals that upper layers exhibit higher diagonality than lower layers as shown in Figure 1, because they concentrate on localized information around the current speech frame to convert phoneme-level information to output text (e.g., "/k/,/ae/,/t/" becomes "cat").

Meanwhile, the conventional CYCLE structure (Takase and Kiyono 2023), which has achieved successful results in recent LLMs, operates by repeatedly stacking a block of independent layers in the same order. When we analyze the

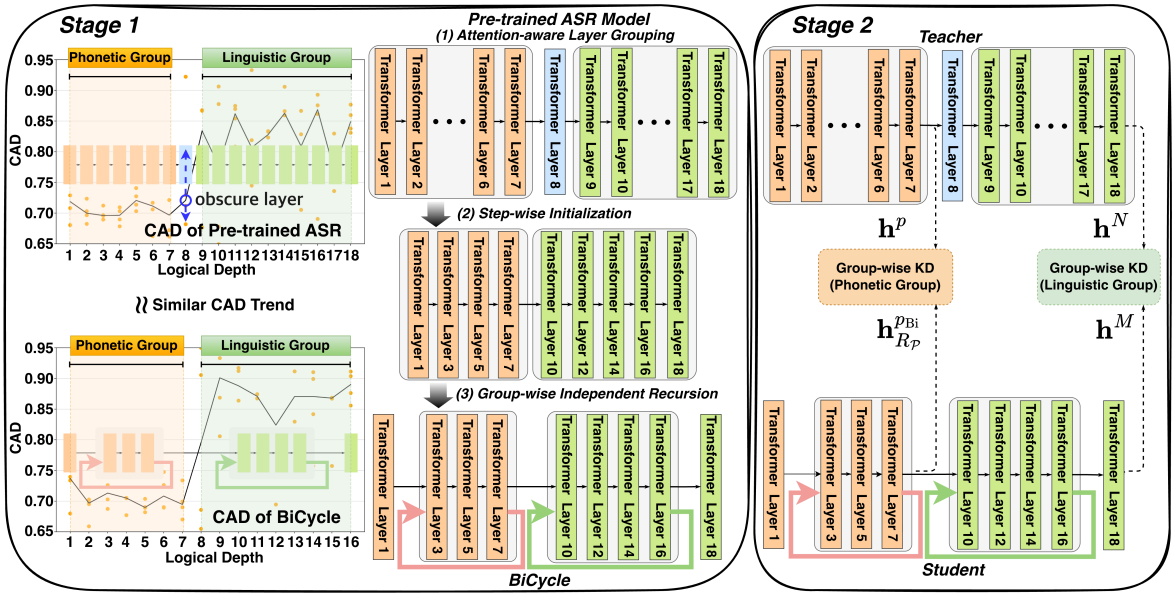


Figure 2: Overview of the BiCycle framework for transformer-based ASR models. In Stage 1, based on cumulative attention diagonality (CAD) analysis, layers are grouped into phonetic (orange), obscure (blue), and linguistic (green) groups to initialize the BiCycle model and construct a recursive structure that preserves the phonetic-to-linguistic pipeline. In Stage 2, by transferring knowledge from the pre-trained ASR model to each group, BiCycle effectively receives knowledge tailored to the role of each individual layer.

CAD of this CYCLE structure, it shows periodic pattern as the entire layer block is considered a single recursive target (Figure 1 (b)). This means that after linguistic processing is completed in the upper layers, the representation from the final layer feeds back into the first layer, leading to periodic repetition of phonetic-to-linguistic processing (e.g. phonetic $\rightarrow$ linguistic $\rightarrow$ phonetic $\rightarrow$ linguistic). This periodic diagonality pattern is a consequence of the recursive structure and fails to reflect the standard ASR mechanism. It suggests that the CYCLE structure does not follow the typical speech recognition process, implying that this RT structure may be sub-optimal for ASR. From this observation, we were motivated to ask: "How can we design an RT structure for ASR that respects the phonetic-to-linguistic processing inherent in speech recognition?"

BiCycle

In this paper, we propose BiCycle, a novel RT scheme specifically designed for transformer-based ASR models. As shown in Figure 2, the proposed framework consists of an initialization step that creates an RT structure considering the speech recognition mechanism, and a distillation step that effectively transfers knowledge from the pre-trained ASR model (teacher) according to layer-wise specialization.

Attention-aware Layer Grouping. To design the RT structure that reflects the typical ASR pipeline, which progresses from phonetic to linguistic processing, we begin by analyzing the attention patterns of the pre-trained ASR model to differentiate the functional roles of encoder layers. As observed, upper layers tend to exhibit more diagonal attention patterns, indicating a stronger focus on localizing linguistic elements for text conversion. In contrast, lower layers show broader receptive fields associated with phonetic representation. Through our CAD-based analysis, we identify an intermediate layer, referred to as the obscure layer, which exhibits ambiguous behavior and lies between phonetic and linguistic processing. In our experiment, this corresponded to the 8th layer.

The ASR mechanism can be described as a hierarchical encoding process that progressively transforms low-level phonetic representations into high-level linguistic abstractions. Let $f(\cdot;\theta_\ell)$ represent the transformation function of the ℓ-th layer, parameterized by θ_ℓ. Given the input feature $\mathbf{h}^0$ and a total of N transformer layers, the hidden representations can be computed recursively as:

$$\mathbf{h}^\ell = f(\mathbf{h}^{\ell-1};\theta_\ell), \quad \ell \in \{1,\dots,N\}.$$

We partition the transformer layers into two functional groups based on their observed behavior. The groups are defined as:

$$\text{Phonetic Group } \mathcal{P} = \{\ell \mid 1 \le \ell \le p\},$$
$$\text{Linguistic Group } \mathcal{K} = \{\ell \mid p+2 \le \ell \le N\}$$

where the obscure layer $\{p+1\}$ is excluded from both groups due to its transitional characteristics.

Step-wise Initialization. After the attention-aware layer grouping, we initialize the BiCycle model by transferring weights from the pre-trained ASR model in a group-wise manner. During this process, only the phonetic and linguistic groups are reused, while the obscure layer is intentionally excluded from initialization due to its transitional nature.

In previous studies (Li et al. 2025a; Sun et al. 2025), the first and last layers in LLMs were shown to play particularly important roles, and we observe a similar pattern in the ASR setting, as will be further discussed in the analysis section. Accordingly, our step-wise initialization explicitly includes the weights of these two layers. In addition, we initialize the phonetic group of the BiCycle model in a bottom-up manner, and the linguistic group in a top-down manner. This enables the construction of a BiCycle model with M layers by selectively transferring weights from the N-layer pre-trained model. Suppose the pre-trained model has p layers in the phonetic group. We initialize the phonetic group of the BiCycle model using only the odd-indexed layers among these p layers, resulting in $\lfloor \frac{p+1}{2} \rfloor$ layers being transferred. Specifically, the BiCycle parameters of the phonetic group $\{\theta_1^{\text{Bi}}, \ldots, \theta_{\lfloor \frac{p+1}{2} \rfloor}^{\text{Bi}}\}$ are initialized as:

$$\theta_i^{\text{Bi}} \leftarrow \theta_{2i-1}, \quad \text{for } i = 1, \ldots, \left\lfloor \frac{p+1}{2} \right\rfloor.$$

For the linguistic group, we initialize the BiCycle layers $\{\theta_{\lfloor \frac{p+1}{2} \rfloor + 1}^{\text{Bi}}, \ldots, \theta_M^{\text{Bi}}\}$ using the odd-indexed upper layers in reverse order:

$$\theta_{M-(j-1)}^{\text{Bi}} \leftarrow \theta_{N-2(j-1)}, \quad \text{for } j = 1, \ldots, M - \lfloor \frac{p+1}{2} \rfloor.$$

This initialization strategy leverages the structural knowledge encoded in the pre-trained model while allowing flexibility in the BiCycle architecture. We intentionally exclude the obscure layer from this process to maintain a clean separation between the phonetic and linguistic groups.

Group-wise Independent Recursion. Through the preceding initialization, each layer is explicitly assigned to either the phonetic or linguistic group. The primary goal of group-wise independent recursion is to leverage this separation to preserve the inherent phonetic-to-linguistic structure of ASR. To this end, recursion is applied independently within each group, rather than across groups. By keeping phonetic and linguistic layers separate during recursion, this strategy prevents interference between distinct processing types and allows each group to perform its specialized role without disruption. The detailed procedure is described in Algorithm 1.

Group-wise Feature Distillation. GFD is a specialized feature-level KD specifically designed for the BiCycle architecture. It enables the BiCycle model (student) to learn from the intermediate representations of the pre-trained ASR model (teacher). Based on the previously identified roles of each layer, the final representations from each of the phonetic and linguistic groups in the teacher model are selectively extracted and distilled into their corresponding groups in the BiCycle model.

Specifically, the student model is initially trained with the GFD loss, followed by additional training using the CTC loss. For each group $g \in \{\mathcal{P}, \mathcal{K}\}$, the difference between the L2-normalized intermediate representations of the student and teacher models is computed using the mean squared error (MSE) loss, yielding the group-wise distillation loss $\mathcal{L}_{GFD,g}$.

Algorithm 1: Group-wise Independent Recursion in BiCycle

Require: Input feature $\mathbf{h}^0$, BiCycle parameters $\{\theta_1^{\text{Bi}}, \ldots, \theta_M^{\text{Bi}}\}$, recursion steps $R_\mathcal{P}, R_\mathcal{K}$, phonetic group size $p_{\text{Bi}} = \lfloor \frac{p+1}{2} \rfloor$.
Notation: $\mathbf{h}_r^\ell$ denotes the hidden state at layer ℓ during the r-th recursion step.
Ensure: Output hidden state $\mathbf{h}^M$

Step 1: Initial layer (non-recursive)
$\mathbf{h}^1 \leftarrow f(\mathbf{h}^0; \theta_1^{\text{Bi}})$
Step 2: Phonetic group recursion
$\mathbf{h}_0^{p_{\text{Bi}}} \leftarrow \mathbf{h}^1$
for $r = 1$ to $R_\mathcal{P}$ **do**
  $\mathbf{h}_r^2 \leftarrow f(\mathbf{h}_{r-1}^{p_{\text{Bi}}}; \theta_2^{\text{Bi}})$
  for $\ell = 3$ to p_{Bi} **do**
    $\mathbf{h}_r^\ell \leftarrow f(\mathbf{h}_r^{\ell-1}; \theta_\ell^{\text{Bi}})$
  end for
end for
$\mathbf{h}_{\text{final}}^{p_{\text{Bi}}} \leftarrow \mathbf{h}_{R_\mathcal{P}}^{p_{\text{Bi}}}$
Step 3: Linguistic group recursion
$\mathbf{h}_0^{M-1} \leftarrow \mathbf{h}_{\text{final}}^{p_{\text{Bi}}}$
for $r = 1$ to $R_\mathcal{K}$ **do**
  $\mathbf{h}_r^{p_{\text{Bi}}+1} \leftarrow f(\mathbf{h}_{r-1}^{M-1}; \theta_{p_{\text{Bi}}+1}^{\text{Bi}})$
  for $\ell = p_{\text{Bi}} + 2$ to $M - 1$ **do**
    $\mathbf{h}_r^\ell \leftarrow f(\mathbf{h}_r^{\ell-1}; \theta_\ell^{\text{Bi}})$
  end for
end for
$\mathbf{h}_{\text{final}}^{M-1} \leftarrow \mathbf{h}_{R_\mathcal{K}}^{M-1}$
Step 4: Final output layer (non-recursive)
$\mathbf{h}^M \leftarrow f(\mathbf{h}_{\text{final}}^{M-1}; \theta_M^{\text{Bi}})$
return $\mathbf{h}^M$

For the phonetic group, the student's representation is the final hidden state from its phonetic recursion, $\mathbf{h}_{R_\mathcal{P}}^{p_{\text{Bi}}}$. The teacher's representation is the hidden state from its final phonetic layer, $\mathbf{h}^p$. The loss is defined as:

$$\mathcal{L}_{GFD,\mathcal{P}} = \text{MSE}\left(\frac{\mathbf{h}_{R_\mathcal{P}}^{p_{\text{Bi}}}}{\|\mathbf{h}_{R_\mathcal{P}}^{p_{\text{Bi}}}\|_2}, \frac{\mathbf{h}^p}{\|\mathbf{h}^p\|_2}\right) \tag{1}$$

For the linguistic group, we align the student's final linguistic layer output $\mathbf{h}^M$ with the teacher's final linguistic layer output $\mathbf{h}^N$. The loss is defined as:

$$\mathcal{L}_{GFD,\mathcal{K}} = \text{MSE}\left(\frac{\mathbf{h}^M}{\|\mathbf{h}^M\|_2}, \frac{\mathbf{h}^N}{\|\mathbf{h}^N\|_2}\right) \tag{2}$$

These individual group-wise losses are summed to form the total GFD loss, $\mathcal{L}_{GFD}$, which is then minimized during model training:

$$\mathcal{L}_{GFD} = \mathcal{L}_{GFD,\mathcal{P}} + \mathcal{L}_{GFD,\mathcal{K}} \tag{3}$$

Traditional feature distillation in ASR often focuses on the final layer (Yoon et al. 2021). However, GFD is a more effective approach because it directly integrates representations from the teacher model, which are aligned with the specific role of each layer, into the loss calculation. This allows for a more comprehensive and effective transfer of knowledge to the student model.

Methods		#Param	Logical Depth	test-clean		test-other		dev-clean		dev-other	
				WER	RERR	WER	RERR	WER	RERR	WER	RERR
Pre-trained Conformer-CTC		30.5M	18	2.96%	–	7.06%	–	2.98%	–	7.03%	–
Conformer-CTC w/o Init (baseline)		16.2M	9	4.53%	–	11.44%	–	4.39%	–	11.62%	–
CYCLE (Takase and Kiyono 2023)		16.2M	18	3.85%	15.01%	9.30%	18.71%	3.67%	16.40%	9.13%	21.43%
			27	4.16%	8.17%	10.23%	10.58%	4.03%	8.20%	10.15%	12.65%
SEQUENCE (Takase and Kiyono 2023)		16.2M	18	4.12%	9.05%	10.02%	12.41%	4.00%	8.88%	9.86%	15.15%
			27	DNC	–	DNC	–	DNC	–	DNC	–
Relaxed Recursive Transformer (Bae et al. 2025)		17.5M	18	7.20%	-58.94%	16.55%	-44.67%	7.15%	-62.87%	7.03%	39.50%
Zero Token Transformer (Li et al. 2025a)		16.5M	16	3.75%	17.22%	9.29%	18.79%	3.66%	16.63%	9.22%	20.65%
			23	3.78%	16.56%	9.07%	20.72%	3.59%	18.22%	9.04%	22.20%
BiCycle (Ours)	$R_{\mathcal{P}}:1, R_{\mathcal{K}}:1$	16.2M	9	3.99%	11.92%	9.88%	13.64%	3.93%	10.48%	9.98%	14.11%
	$R_{\mathcal{P}}:2, R_{\mathcal{K}}:2$	16.2M	16	3.79%	16.34%	9.11%	20.37%	3.51%	20.05%	9.20%	20.83%
	$R_{\mathcal{P}}:2, R_{\mathcal{K}}:3$		20	**3.67%**	**18.98%**	**8.93%**	**21.94%**	3.49%	20.50%	**8.98%**	**22.72%**
	$R_{\mathcal{P}}:3, R_{\mathcal{K}}:2$	16.2M	19	3.71%	18.10%	9.06%	20.80%	3.56%	18.91%	9.07%	21.94%
	$R_{\mathcal{P}}:3, R_{\mathcal{K}}:3$		23	3.83%	15.45%	9.01%	21.24%	**3.46%**	**21.18%**	9.07%	21.94%

Table 1: Comparison of WER and RERR with existing RT frameworks on the LibriSpeech. DNC denotes "Did Not Converge".

Methods		#Param	Logical Depth	test-clean		test-other		dev-clean		dev-other	
				WER	RERR	WER	RERR	WER	RERR	WER	RERR
Pre-trained Conformer-CTC		30.5M	18	2.96%	–	7.06%	–	2.98%	–	7.03%	–
KD	Conformer-CTC (baseline)	16.2M	9	4.47%	–	11.34%	–	4.33%	–	11.36%	–
	CYCLE (Takase and Kiyono 2023)	16.2M	18	3.72%	16.78%	8.89%	21.60%	3.46%	20.09%	**8.85%**	**22.10%**
	Zero Token Transformer (Li et al. 2025a)	16.5M	16	3.64%	18.57%	9.16%	19.22%	3.47%	19.86%	8.96%	21.13%
	BiCycle (Ours)	16.2M	16	**3.57%**	**20.13%**	**8.64%**	**23.81%**	**3.38%**	**21.94%**	8.86%	22.01%

Table 2: Comparison of WER and RERR with existing RT frameworks using KD on the LibriSpeech benchmark.

Experiments

Experimental Setup

To evaluate our proposed RT, we assessed the model's performance using two datasets: LibriSpeech (Panayotov et al. 2015) and Common Voice 7.0 (Ardila et al. 2020) French. For evaluation, we employed two widely used metrics: word error rate (WER) and relative error rate reduction (RERR), where RERR quantifies the proportional decrease in WER compared to the baseline. All experiments were implemented using the NeMo toolkit (Kuchaiev et al. 2019), and greedy decoding was applied during inference. As a baseline, we employed a conformer (Gulati et al. 2020)-CTC (Graves et al. 2006) model. Specifically, we used 30M- and 121M-parameter conformer-CTC models as the pre-trained ASR backbones for LibriSpeech and Common Voice 7.0 French, respectively, and applied RT methods to these baselines. We applied CYCLE, SEQUENCE (Takase and Kiyono 2023), Relaxed Recursive Transformer (Bae et al. 2025), and Zero Token Transformer (Li et al. 2025a) as the competing RT methods. Additionally, we observed that Foldable Network (Li et al. 2025b) failed to converge when initialized with pre-trained weights, although it was able to learn when trained from scratch. However, since all other methods in our comparison leveraged transferred weights for initialization, Foldable Network's overall performance remained substantially lower than other methods that used transferred initialization. Due to this large performance gap, we excluded it from our final comparison. Though various initialization strategies for RT exist, conventional RT methods were initialized using our proposed step-wise strategy, as it yielded the best empirical performance. For Relaxed Recursive Transformer, we followed the original initialization strategy based on a weighted average. All RT variants were trained for 100 epochs using the CTC loss. For the feature distillation setting, models were first trained for 30 epochs with distillation loss, followed by 100 additional epochs using the CTC loss. Detailed implementation details will be provided in Appendix.

Experimental Results

Main Results. Table 1 reports the WER and RERR results on the LibriSpeech benchmark. We used a CTC model without both initialization and recursion as the baseline for computing RERR. From the results, it is confirmed that conventional methods such as CYCLE, SEQUENCE, and

Methods	#Param	Logical Depth	test-clean		test-other		dev-clean		dev-other	
			WER	RERR	WER	RERR	WER	RERR	WER	RERR
Pre-trained Conformer-CTC	30.5M	18	2.96%	–	7.06%	–	2.98%	–	7.03%	–
Conformer-CTC (baseline)	16.2M	9	4.53%	–	11.44%	–	4.39%	–	11.62%	–
FitNets (Romero et al. 2015) + BiCycle	16.2M	20	3.56%	21.41%	8.51%	25.61%	3.38%	23.01%	8.44%	27.37%
GFD + BiCycle (Ours)			**3.38%**	**25.39%**	8.43%	26.31%	3.26%	25.74%	8.45%	27.28%
FitNets (Romero et al. 2015) + BiCycle	16.2M	23	3.50%	22.74%	8.53%	25.44%	3.31%	24.60%	8.40%	27.71%
GFD + BiCycle (Ours)			3.53%	22.08%	**8.14%**	**28.85%**	**3.25%**	**25.97%**	**8.31%**	**28.49%**

Table 3: Comparison of WER and RERR with conventional feature-level KD on the LibriSpeech benchmark.

Methods	#Param	Logical Depth	test		dev	
			WER	RERR	WER	RERR
Pre-trained Conformer-CTC	121M	18	11.71%	–	10.66%	–
Conformer-CTC (baseline)	64.6M	9	14.99%	–	13.15%	–
CYCLE (Takase and Kiyono 2023)	64.6M	18	12.55%	16.28%	11.06%	15.89%
Zero Token Transformer (Li et al. 2025a)	65.8M	16	12.51%	16.54%	11.17%	15.06%
BiCycle w/o Recursion (Ours)	64.6M	9	13.16%	12.21%	11.64%	11.48%
BiCycle (Ours)		16	**12.38%**	**17.41%**	**10.89%**	**17.19%**

Table 4: Comparison of WER and RERR with existing RT frameworks on the CommonVoice 7.0 French benchmark.

Relaxed Recursive Transformer, which were originally designed for LLMs, yielded relatively limited performance improvements in the ASR setting. Moreover, these methods did not benefit from increased recursion steps, which further undermined the practical utility of RT in ASR tasks. In contrast, BiCycle consistently delivered substantial performance gains across all evaluation datasets, achieving the best WER performance in most configurations. At a logical depth of 16, BiCycle achieved a 20.40 % RERR on test-other compared to the baseline, a particularly notable result that highlighted the effectiveness of its recursive design. Furthermore, unlike other RT methods, BiCycle benefited from deeper recursion, achieving its highest performance at greater logical depths (e.g., 20). This confirms its ability to leverage increased recursion depth to enhance performance.

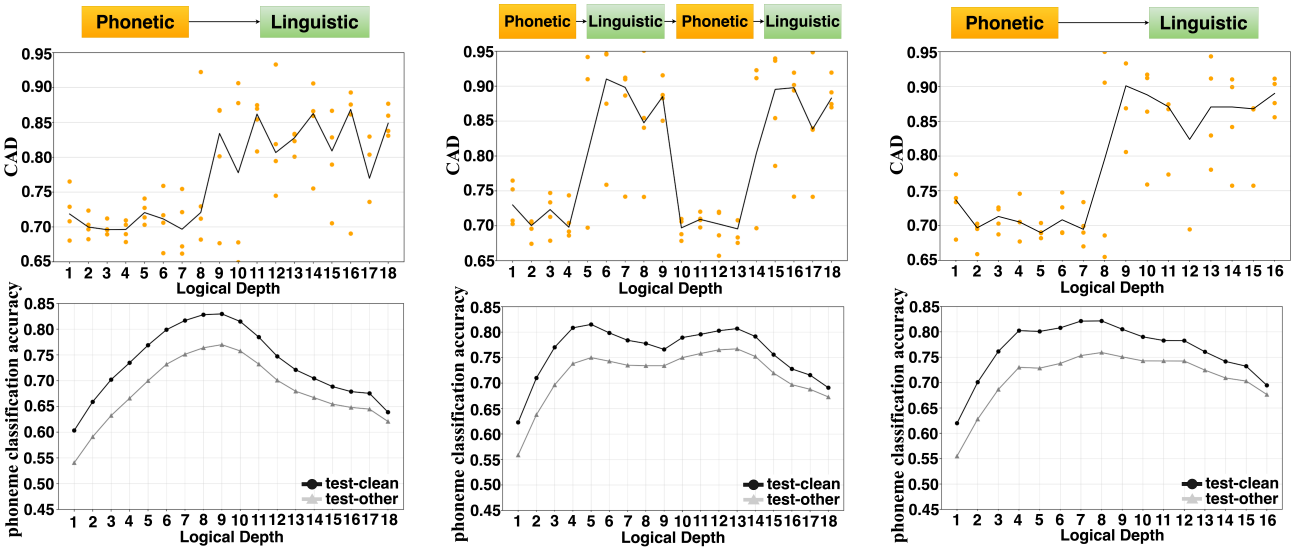
Performance with KD. Table 2 presents the results of applying KD. We selected conventional RT methods that showed notable improvements over the baseline in the previous experiment. Specifically, KD was performed using KL divergence, where the soft targets were obtained from the output distribution of the pre-trained ASR model. From the results, it is verified that BiCycle maintained its superior performance, achieving the best results in most scenarios even after applying KD, with the only exception being the dev-other set, where it fell short by just 0.01.

Effectiveness of GFD. Building on this finding, we further enhanced BiCycle's performance by incorporating feature-level KD. As previously mentioned, conventional feature-level KD in ASR typically relied solely on the final hidden representation of the pre-trained teacher model (Yoon et al. 2021), following the FitNets paradigm (Romero et al. 2015). In contrast, GFD performed group-wise feature distillation between the teacher and student models' phonetic and linguistic groups, effectively transferring knowledge aligned with each group's functional role. Table 3 compares GFD with conventional feature-level KD, which used only the final hidden representations of the teacher and student models. The results show that GFD performed better than conventional method, confirming the effectiveness of group-wise knowledge transfer in the BiCycle architecture.

Results for French Dataset. While LibriSpeech is the most widely-used benchmark in speech recognition, we believe that using additional language benchmark can enhance the generalizability of the proposed method. Consequently, we conducted new experiments on the Common Voice 7.0 French dataset, evaluating the model on both the dev and test As shown in Table 4, our RT model achieved the best performance across all evaluation metrics. These results further supported the effectiveness of modeling ASR's inherent phonetic-to-linguistic processing structure, validating the core design principle of our approach.

Analysis

Speech Recognition Process in BiCycle. As shown in Figure 3, we analyzed layer-wise changes in CAD and phoneme classification accuracy (Shim, Choi, and Sung 2022) for three models: a pre-trained ASR model, the conventional CYCLE structure, and our proposed BiCycle architecture. The phoneme classification accuracy of the



(a) Standard ASR Model (b) CYCLE (c) BiCycle (Ours)

Figure 3: Graph of CAD and phoneme classification performance on LibriSpeech test datasets according to model architecture.

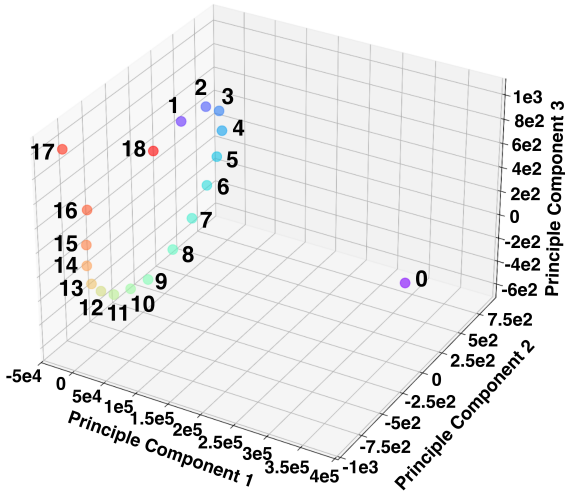


Figure 4: 3D visualization of layer-wise representations using PCA. The first (0) and last (18) layers are notably distant from intermediate layers.

pre-trained model consistently increased in the lower layers but showed a decreasing trend in the upper layers due to linguistic localization. This pattern of accuracy change aligned with the CAD pattern, which exhibited low diagonality in the lower layers and high diagonality in the upper layers. As previously analyzed, the CYCLE structure exhibited a periodic CAD pattern, which was similarly reflected in its phoneme classification accuracy. A periodic pattern was observed where the accuracy temporarily dropped due to linguistic localization, but then rised again as the recursion repeats the phonological processing. In contrast, our proposed BiCycle structure was designed to directly reflect the processing flow of the standard ASR model, which performed phonetic standardization followed by text conversion. Consequently, BiCycle's CAD results showed a diagonal pattern identical to that of the pre-trained model. Its phoneme classification accuracy also maintained the same trend as the original model: continuously rising during the phonetic standardization phase and then decreasing during the linguistic localization stage. This demonstrates that BiCycle successfully preserved the ASR processing mechanism, even while utilizing the recursive structure.

Head-Tail Decoupling. To analyze why ASR model performs better when we separate the first and last layers from the recurrent structure, we used principal component analysis (PCA). We visualized the intermediate representations after passing through the layers as a 3D scatter plot. The distance between points visually represented how structurally different those data points were in the original high-dimensional space. As Figure 4 illustrates, the representations after the first layer and the last layer were significantly distant from the representations of the preceding layers. This indicates that these layers acquired very different types of information as the data passed through them. Therefore, by removing the first and last layers from the recurrent structure, we prevented functional conflicts that could arise within the recurrent block.

Conclusion

In this paper, we introduced BiCycle, a novel RT scheme for transformer-based ASR models. BiCycle effectively mitigated conflicts between phonetic and linguistic processing by structurally separating them within its recursive architecture. We also proposed GFD, which distilled knowledge for these distinct processing groups individually. Our empirical results showed that the proposed method significantly improved ASR performance over conventional RT approaches.

Acknowledgements

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2025-00515722). This work was also supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [RS-2021-II211341, Artificial Intelligence Graduate School Program (Chung-Ang University)].

References

Ardila, R.; Branson, M.; Davis, K.; Henretty, M.; Kohler, M.; Meyer, J.; Morais, R.; Saunders, L.; Tyers, F. M.; and Weber, G. 2020. Common voice: A massively-multilingual speech corpus. *In Proc. LREC*.

Bae, S.; Fisch, A.; Harutyunyan, H.; Ji, Z.; Kim, S.; and Schuster, T. 2025. Relaxed recursive transformers: Effective parameter sharing with layer-wise lora. *In Proc. ICLR*.

Baevski, A.; Zhou, Y.; Mohamed, A.; and Auli, M. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *In Proc. NeurIPS*.

Bahri, Y.; Dyer, E.; Kaplan, J.; Lee, J.; and Sharma, U. 2021. Explaining neural scaling laws. *arXiv preprint arXiv:2502.12214*.

Cao, Z.; Yang, Y.; and Zhao, H. 2024. Head-wise shareable attention for large language models. *arXiv preprint arXiv:2402.11819*.

Chen, W.; Tian, J.; Peng, Y.; Yan, B.; Yang, C.-H. H.; and Watanabe, S. 2025. OWLS: Scaling laws for multilingual speech recognition and translation models. *arXiv preprint arXiv:2502.10373*.

Graves, A.; Fernández, S.; Gomez, F.; and Schmidhuber, J. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. *In Proc. ICML*.

Gu, Y.; Shivakumar, P. G.; Kolehmainen, J.; Gandhe, A.; Rastrow, A.; and Bulyko, I. 2023. Scaling laws for discriminative speech recognition rescoring models. *arXiv preprint arXiv:2306.15815*.

Gulati, A.; Qin, J.; Chiu, C.-C.; Parmar, N.; Zhang, Y.; Yu, J.; Han, W.; Wang, S.; Zhang, Z.; Wu, Y.; et al. 2020. Conformer: Convolution-augmented transformer for speech recognition. *In Proc. INTERSPEECH*.

Hinton, G.; Vinyals, O.; and Dean, J. 2014. Distilling the knowledge in a neural network. *In Proc. NIPS Workshop Deep Learn*.

Hsu, W.-N.; Bolte, B.; Tsai, Y.-H. H.; Lakhotia, K.; Salakhutdinov, R.; and Mohamed, A. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*, 29: 3451–3460.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2022. Lora: Low-rank adaptation of large language models. *In Proc. ICLR*.

Kaplan, J.; McCandlish, S.; Henighan, T.; Brown, T. B.; Chess, B.; Child, R.; Gray, S.; Radford, A.; Wu, J.; and Amodei, D. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Kashiwagi, Y.; Futami, H.; Tsunoo, E.; and Asakawa, S. 2025. Whale: Large-Scale multilingual ASR model with w2v-BERT and E-Branchformer with large speech data. *arXiv preprint arXiv:2506.01439*.

Kuchaiev, O.; Li, J.; Nguyen, H.; Hrinchuk, O.; Leary, R.; Ginsburg, B.; Kriman, S.; Beliaev, S.; Lavrukhin, V.; Cook, J.; et al. 2019. Nemo: a toolkit for building ai applications using neural modules. *arXiv preprint arXiv:1909.09577*.

Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; and Soricut, R. 2020. Albert: A lite bert for self-supervised learning of language representations. *In Proc. ICLR*.

Li, G.; Jiang, W.; Shen, L.; Tang, M.; and Yuan, C. 2025a. Zero token-driven deep thinking in llms: Unlocking the full potential of existing parameters via cyclic refinement. *arXiv preprint arXiv:2502.12214*.

Li, Z.; Xu, H.; Xie, X.; Jin, Z.; Wang, T.; and Liu, X. 2025b. Unfolding A Few Structures for The Many: Memory-Efficient Compression of Conformer and Speech Foundation Models. *In Proc. INTERSPEECH*.

Liu, A.; Feng, B.; Xue, B.; Wang, B.; Wu, B.; Lu, C.; Zhao, C.; Deng, C.; Zhang, C.; Ruan, C.; et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Minaee, S.; Mikolov, T.; Nikzad, N.; Chenaghlu, M.; Socher, R.; Amatriain, X.; and Gao, J. 2024. Large language models: A survey. *arXiv preprint arXiv:2402.06196*.

Nouriborji, M.; Rohanian, O.; Kouchaki, S.; and Clifton, D. A. 2023. Minialbert: model distillation via parameter-efficient recursive transformers. *In Proc. ACL Anthology*.

Panayotov, V.; Chen, G.; Povey, D.; and Khudanpur, S. 2015. Librispeech: an asr corpus based on public domain audio books. *In Proc. ICASSP*, 5206–5210.

Romero, A.; Ballas, N.; Kahou, S. E.; Chassang, A.; Gatta, C.; and Bengio, Y. 2015. FitNets: Hints for Thin Deep Nets. *In Proc. ICLR*.

Shen, Z.; Liu, Z.; and Xing, E. 2022. Sliced recursive transformer. *In Proc. Springer*, 727–744.

Shim, K.; Choi, J.; and Sung, W. 2022. Understanding the role of self attention for efficient speech recognition. *In Proc. ICLR*.

Song, X.; Wu, D.; Zhang, B.; Zhou, D.; Peng, Z.; Dang, B.; Pan, F.; and Yang, C. 2024. U2++ moe: Scaling 4.7 x parameters with minimal impact on rtf. *arXiv preprint arXiv:2404.16407*.

Sun, Q.; Pickett, M.; Nain, A. K.; and Jones, L. 2025. Transformer layers as painters. *In Proc. AAAI*.

Takase, S.; and Kiyono, S. 2023. Lessons on parameter sharing across layers in transformers. *In Proc. SustaiNLP*.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *In Proc. NeurIPS*.

Wang, J.; Chen, Y.-G.; Lin, I.-C.; Li, B.; and Zhang, G. L. 2025. Basis sharing: Cross-layer parameter sharing for large language model compression. *In Proc. ICLR*.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yang, S.-w.; Liu, A. T.; and Lee, H.-y. 2020. Understanding self-attention of self-supervised audio transformers. *In Proc. INTERSPEECH*.

Yoon, J. W.; Lee, H.; Kim, H. Y.; Cho, W. I.; and Kim, N. S. 2021. Tutornet: Towards flexible knowledge distillation for end-to-end speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29: 1626–1638.

Zhai, X.; Kolesnikov, A.; Houlsby, N.; and Beyer, L. 2022. Scaling vision transformers. *In Proc. CVPR*, 12104–12113.

Zhang, K.; Chkhetiani, L.; Ramirez, F. M.; Khare, Y.; Vanzo, A.; Liang, M.; Martin, S. R.; Oexle, G.; Bousbib, R.; Peyash, T.; et al. 2024. Conformer-1: Robust asr via large-scale semisupervised bootstrapping. *arXiv preprint arXiv:2404.07341*.