RingFormer: Rethinking Recurrent Transformer with Adaptive Level Signals

Anonymous ACL submission

Abstract

Transformers have achieved great success in effectively processing sequential data such as text. Their architecture consisting of several attention and feedforward blocks can model relations between elements of a sequence in parallel manner, which makes them very efficient to train and effective in sequence modeling. Even though they have shown strong performance in processing sequential data, the size of their parameters is considerably larger when compared to other architectures such as RNN and CNN based models. Therefore, several approaches have explored parameter sharing and recurrence in Transformer models to address their computational demands. However, such methods struggle to maintain high performance compared to the original Transformer model. To address this challenge, we propose our novel approach, *RingFormer*, which employs one Transformer layer that processes input repeatedly in a circular, ring-like manner, while utilizing low-rank matrices to generate input-dependent level signals. This allows us to reduce the model parameters substantially while maintaining high performance in a variety of tasks such as translation and image classification, as validated in the experiments.

1 Introduction

Transformer models, since their introduction (Vaswani et al., 2017), have dramatically transformed the landscape of deep learning, particularly excelling in tasks involving sequential data such as natural language processing (Brown et al., 2020; Radford et al., 2019) and machine translation (Ott et al., 2018). Not long after their inception, they have also shown strong performance in various other domains such as reinforcement learning (Chen et al., 2021), image classification (Dehghani et al., 2023; Dosovitskiy et al., 2020; Liu et al., 2021), object detection (Carion et al., 2020) and image generation (Jiang et al., 2021; Peebles and Xie, 2022; Zhang et al., 2022). Their core architecture, characterized by self-attention mechanisms and feedforward neural networks, enables effective handling of long-range dependencies and parallel processing of input sequences. The ability of this architecture to model intricate relationships within data has led to significant breakthroughs, making it a foundation model across many modern large-scale AI systems (Anthropic, 2023; Google, 2024; OpenAI et al., 2024; Touvron et al., 2023).

However, the impressive capabilities of transformer models come with substantial computational and memory costs (Brown et al., 2020; Dosovitskiy et al., 2020). The standard Transformer architecture consists of multiple layers, each containing millions of parameters that need to be trained and stored. This results in high memory usage and significant computational demands, often requiring specialized hardware. Moreover, deploying these models in resource-constrained environments, such as mobile devices or edge computing scenarios, becomes challenging due to their size and complexity. These limitations have spurred a growing interest in developing more parameter-efficient Transformer architectures (Dehghani et al., 2019; Pires et al., 2023) that can retain their powerful performance while being less resource intensive.

In this paper, we introduce a Transformer architecture that recurrently leverages a single shared Transformer block in a novel way by integrating input-dependent level signals at each block iteration, which are shown to be crucial for adapting the shared block to different stages of the model. The level signals are generated by depth-specific low-rank transformations applied to the input in the attention and feedforward layers within the Transformer block. Our *RingFormer* model can also be viewed as stacking Transformer layers whose parameters combine (1) a set of global parameters shared across all Transformer layers and (2) a set of local low-rank layer-dependent parameters. This simple design effectively addresses the trade-off between reducing the number of model parameters and limiting the model's capacity to capture complex patterns.

We validate our model through experiments and analysis on machine translation and image classification. The results of experiments and analysis demonstrate that our model closely replicates the behavior of the original Transformer model, and it performs better against existing parameter-matched recurrence-based Transformer models, underscoring the effectiveness of our approach in maintaining high performance with fewer parameters.

The contributions of this paper are summarized as follows:

- We enhance a recurrent Transformer architecture to significantly reduce the model's parameter count while maintaining high performance.
- We propose novel input-dependent level signals generated in a parameter-efficient way using low-rank matrices to improve the adaptability of a recurrent Transformer model, and show that those signals help the model replicate the behavior of the original model.
- We demonstrate the validity of our approach through careful analysis and ablation studies, and show the effectiveness of our model on tasks such as translation and image classification.

2 Background

2.1 Transformer Architecture

The Transformer architecture (Vaswani et al., 2017) comprises multiple layers of the same structure stacked together, with each layer consisting of two main modules: *Attention* and *Feedforward Network* described in Equations (1) and (2), respectively. Each of these modules is accompanied by residual connections and layer normalization. In addition, to provide information about the position of tokens in the sequence, the Transformer model adds static sinusoidal or learnable positional encodings to the input embeddings. These encodings allow the model to capture the order within a sequence. The following equations describe the mechanism of two main modules:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d}})V \quad (1)$$

$$FFN(x) = \sigma(xW_{up} + b_{up})W_{down} + b_{down} \quad (2)$$

Here, Q, K, and V are the results of projecting the input vectors through their respective matrices. Attention module can be classified into self-attention (when the Q, K, and V input vectors are the same) and cross-attention (when the Q input vector is different from the K and V input vectors), while the feedforward block consists of up-projection and down-projection transformations with non-linearity function σ between them.

It is well known that Transformer architecture follows a scaling law for both vision tasks and NLP tasks (Dehghani et al., 2023; Hoffmann et al., 2022). This scaling law demonstrates that the performance of Transformer models improves predictably as the model size and computational resources increase. Due to the steep slope of the scaling law, the parameter sizes of Transformer models have continued to grow, leading to significant advancements in their capabilities. However, this growth has also made training and using such massive models increasingly infeasible without substantial GPU resources.

2.2 Related Work

To address the challenge of requiring extensive hardware resources for large Transformer models, researchers have explored various methods to enhance efficiency.

One approach is related to pruning of Transformer model layers, which involves removing less important layers or weights to streamline the model. It was found that many deep layers in large language models are redundant (Gromov et al., 2024), and by pruning up to half of these layers, it was possible to significantly reduce the model size with minimal accuracy degradation.

Another strategy is sharing parameters across different layers or components in Transformers, reducing the model's complexity and memory usage. The Universal Transformers (Dehghani et al., 2019) introduces a model where parameters are shared across layers using a recurrent mechanism with layer-dependent positional encoding, which maintains good performance in various NLP tasks while reducing the number of parameters. People have also proposed sequence and cycle strategies for sharing parameters across layers (Takase and Kiyono, 2021), improving efficiency and performance in tasks like machine translation and speech recognition. Similarly, Subformer (Reid et al., 2021) and One Wide Feedforward (Pires et al., 2023) investigate partial weight sharing within layers, showing that significant parameter reductions can be achieved with little accuracy sacrifice. These models demonstrate that shared parameters can lead to efficient and effective Transformer architectures.

To investigate recurrence-based models, we performed a layer representation similarity analysis using the common CKA (centered kernel alignment) (Kornblith et al., 2019) method and mean attention distance (MAD) (Dosovitskiy et al., 2020) analysis, and we found that the layer representations and internal attention behavior of the previously proposed fully recurrence-based Transformer model (Dehghani et al., 2019) are considerably different compared to those of the original Transformer model.

We hypothesized that the difference in model behavior, especially in attention module, might be the main cause for the gap in performance, and if we can simulate the behavior of the original model using a recurrent model with adaptive level signals, we can also maintain higher performance. Our proposed methodology is focused on addressing this difference, narrowing the gap of the model behavior, and in turn the model performance.

3 Method

3.1 Overview

In this section, we provide a detailed explanation of our proposed work, covering the specific details about the structure of our model, which is illustrated in Figure 1.

The encoder or decoder Transformer-based models consist of several layers with the same structure, where each layer is a combination of sub-layers such as attention and feedforward layers. Those models can be formulated in the following way:

$$\begin{aligned} F(x) &= f_N(f_{N-1}(...f_2(f_1(x)))) \\ &= f(f(...f(f(x,p_1),p_2)),p_{N-1}),p_N) \end{aligned} \tag{3}$$

where N, F, f, x and p_i denote the number of layers, entire encoder (or decoder), each encoder (or decoder) block, input and parameters of each i^{th} layer, respectively. The general formulation of the recurrent Transformer model with level transition functions can be written as below:

$$\begin{aligned} F(x) &= f_N(f_{N-1}(...f_1(x))) \\ f_i(x) &= f_r(x, g_i(x)) \end{aligned} \tag{4}$$

where f_r denotes the recurrent Transformer block and $g_i(x)$ represents a generic level transition function specific for each level. In Universal Transformers (Dehghani et al., 2019), it was shown that using static spatio-temporal positional embeddings can serve as level transition functions for the recurrent Transformer layer and have good model performance. Specifically, in that work, level transition function $g_i(x)$ can be represented as $g_i(x) = x + l(i, x_p)$, where l is a function that returns a positional embedding vector based on level depth i and the position x_p of the vector x, while the i^{th} Transformer block function $f_i(x)$ can be represented as $f_i(x) = f_r(g_i(x))$.

Below, we describe our way of constructing and integrating level transition function $g_i(x)$ to generate adaptive level signals.

3.2 Adaptive Level Signals

To have effective transition between the levels when using recurrent Transformer block, we make $g_i(x)$ directly dependent on the input in the following way: $g_i(x) = M_i \cdot x$, where M is a learnable transformation matrix. Since the main role of level signals is to nudge the input vectors in the right direction, which is an easier task compared to the main input transformation done by the recurrent layer, we hypothesize that making the M matrix low-rank while keeping the recurrent layers at full-rank will let us have parameter-efficiency and high performance at the same time. We draw inspiration for such a low-rank matrix construction and its weight initialization from the parameter efficient fine-tuning (PEFT) technique, LoRA (Hu et al., 2021), and decompose M_i into two low-dimensional matrices, A_i and B_i described in Equation 5.

$$M_i = A_i \cdot B_i^T, \ A_i, B_i \in \mathbb{R}^{d \times r} \text{ and } r \ll d \tag{5}$$

We initialize the A matrices, down-projection parameter matrices, in level signals with zeros, which results in those signals gradually contributing for level adaptation and having stable training. We found that using non-zero (random) initializations could lead to less stable training dynamics, likely due to premature influence of level signals before meaningful representations are formed.

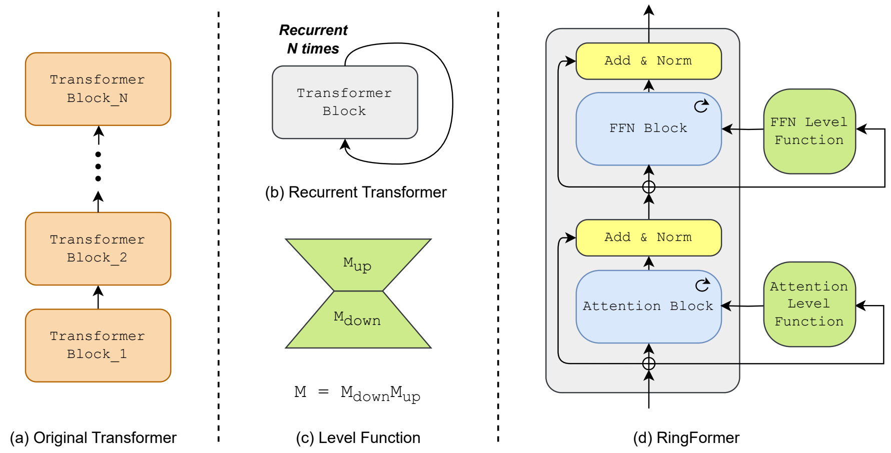

Figure 1: Overview of (a) vanilla Transformer (Vaswani et al., 2017), (b) recurrent Transformer (Dehghani et al., 2019) and (d) our RingFormer architecture. The Transformer block represents either encoder or decoder. In the RingFormer model, a single Transformer block is itself iterated N times, with each sub-modules having unique layer normalization and level signals. Each iteration of the shared block includes both attention and feedforward sub-layers (along with residual connection and normalization), similar to a standard Transformer layer. (c) illustration of the low-rank matrices representing the level functions, where M_{down} down-projects the input to a lower dimensional space and M_{up} up-projects back to the original space.

Since a Transformer layer consists of an attention block and a feedforward block, we generate two distinct signals $g_{Ai}(x)$ and $g_{Fi}(x)$: one for the attention block and the other for the feedforward block, respectively. Additionally, since the Transformer block also has layer normalization applied between the sub-layers, for each level, we allocate unique layer normalization in the attention and feedforward layers. This provides extra input adaptation while only slightly increasing the number of total parameters in the model.

3.2.1 Attention Block

For the attention mechanism, which calculates relevance between elements of a sequence using three projection matrices (query, key, and value), we generate level signals for each of those projections using separate low-rank matrices. We integrate signals after the projection of the input vector x by W_Q, W_K and W_V matrices (shared across the levels) in the following way:

$$Q_i = W_Q \cdot x + g_{A_{Qi}}(x),$$
$$K_i = W_K \cdot x + g_{A_{Ki}}(x), \tag{6}$$
$$V_i = W_V \cdot x + g_{A_{Vi}}(x),$$

where $g_{A_{Qi}}(x) = M_{Qi} \cdot x$, $g_{A_{Ki}}(x) = M_{Ki} \cdot x$, $g_{A_{Vi}}(x) = M_{Vi} \cdot x$. By incorporating the level functions separately for Q, K, and V, we enable fine-grained control over depth-dependent modifications to each component of the attention mechanism. Also, adding the level signals in this manner avoids direct input changes to the main recurrent projections, which was found to be beneficial in our experiments. This can be because such a direct input change can interfere with the learning process of the recurrent layers in the attention module, which needs to solely focus on modeling effective communication between tokens.

3.2.2 Feedforward Block

For feedforward block, the projection of input to intermediate vector of this module requires relatively large number of parameters. Furthermore, there have been various explorations regarding the role of feedforward network in Transformers. One such study (Geva et al., 2021) argues that the feedforward network can be interpreted as a key-value memory pair, where the matrix of the first linear layer is involved in the coefficients of input factors, and the matrix of the second linear layer relates to information about the training corpus. Considering parameter-efficiency and the previous finding, in our approach, for the feedforward network, we add signals before projecting the input using the up-projection layer to guide the coefficient formation of the input in the following way:

$$FFN(x) = \sigma((x + g_{Fi}(x))W_{up})W_{down} \tag{7}$$

where $g_{Fi}(x) = M_{Fi} \cdot x$, the function σ is a non-linear function such as GELU (Hendrycks and Gimpel, 2023), and the bias terms were omitted for brevity.

In encoder-decoder models, we reuse a single Transformer block with attention and feedforward layers in the encoder, while the decoder shares a separate block with cross-attention. We omit level signals in the decoder's cross-attention, as they showed no benefit during development, likely because the cross-attention takes on the output of the already level-adapted attention module as queries, while keys and values come from the encoder's final level. Hence, we prioritized simplicity and avoided redundant overhead in cross-attention.

4 Experiments

We evaluate RingFormer and baseline models on two tasks: WMT-14 German-English translation (Bojar et al., 2014) and ImageNet-based classification (Deng et al., 2009). These benchmarks are widely used, large enough for reliable performance, and feasible given our computational constraints.

As RingFormer uses recursive parameter sharing, we compare it to models with similar strategies. Due to limited research in this area, we selected two representative methods: Universal Transformer (Dehghani et al., 2019) (fully recurrent) and One Wide Feedforward (Pires et al., 2023) (partially reccurent, with shared feedforward layer). These models capture the main variants of parameter sharing and offer meaningful comparisons to RingFormer and the standard Transformer (Vaswani et al., 2017).

4.1 Experimental Details

In this section, we describe each downstream task in detail to support reproducibility. For all models, the rank of the level signal decomposition is set to the input hidden dimension divided by 16. Ablation results for different ranks are shown in Table 4.

Translation As the Transformer was originally proposed for translation (Vaswani et al., 2017), we evaluate our model on WMT-14 German-English (Bojar et al., 2014), which has 4.5M sentence pairs. We report BLEU scores (Papineni et al., 2002) on the test set using a bilingual tokenizer with a 52K BiBERT vocabulary (Xu et al., 2021). Two model sizes are evaluated (see Table 1), trained for 830K steps with a batch size of 512 and 6 layers on two A100 80GB GPUs. We use Adam (Kingma and Ba, 2017) with a cosine scheduler and 40K warm-up steps, and GELU (Hendrycks and Gimpel, 2023) as the activation function.

Table 1 shows model hyperparameters and results. Parameter counts exclude encoder, decoder, and vocabulary head components, as they are fixed per model size. For base models, encoder/decoder embeddings have 26.62M parameters, and the vocabulary head has 26.67M. For large models, these are 53.24M and 53.30M, respectively.

Image Classification As the ViT (Dosovitskiy et al., 2020) model became very prevalent in the vision domain, especially in image classification, we decided to test our model and other baseline models on this task. All models use encoder-only architectures that take image patches with a class token and predict using the final hidden state of that token. We follow the original ViT setup, use sinusoidal spatio-temporal embeddings in Universal Transformer (Dehghani et al., 2019), and share only the feedforward layer in One Wide Feedforward (Pires et al., 2023), keeping attention layers distinct.

We first train smaller models on a subset of the original ImageNet-1K dataset (Deng et al., 2009) for 100 epochs. We randomly chose 100 classes with the total number of 100K training samples (1K per each class) from the original training set, and 5K testing samples (50 per each class) from the original validation set. For easy referencing, we call that subset *ImageNet-small*. As the size of the dataset is relatively small, we decided to train models having only 6 layers / iterations (in the case of recurrent models, we say iterations or levels instead of layers). For bigger size models with 12 layers / iterations, we trained on the whole *ImageNet-1K* for 50 epochs due to limited resources.

The additional training and *ImageNet-small* dataset details are given in Appendix A.1 and A.3. The model hyperparameters, parameter size and experiment results on *ImageNet-small* and *ImageNet-1K* are given in Table 2 and 3.

4.2 Experimental Results

Translation The details of experimental results on translation are presented in Table 1. Our RingFormer model achieves competitive performance with Vanilla Transformer model (Vaswani et al., 2017) and One Wide FFN model (Pires et al., 2023) with less number of parameters for base and large size models. RingFormer outperforms Universal

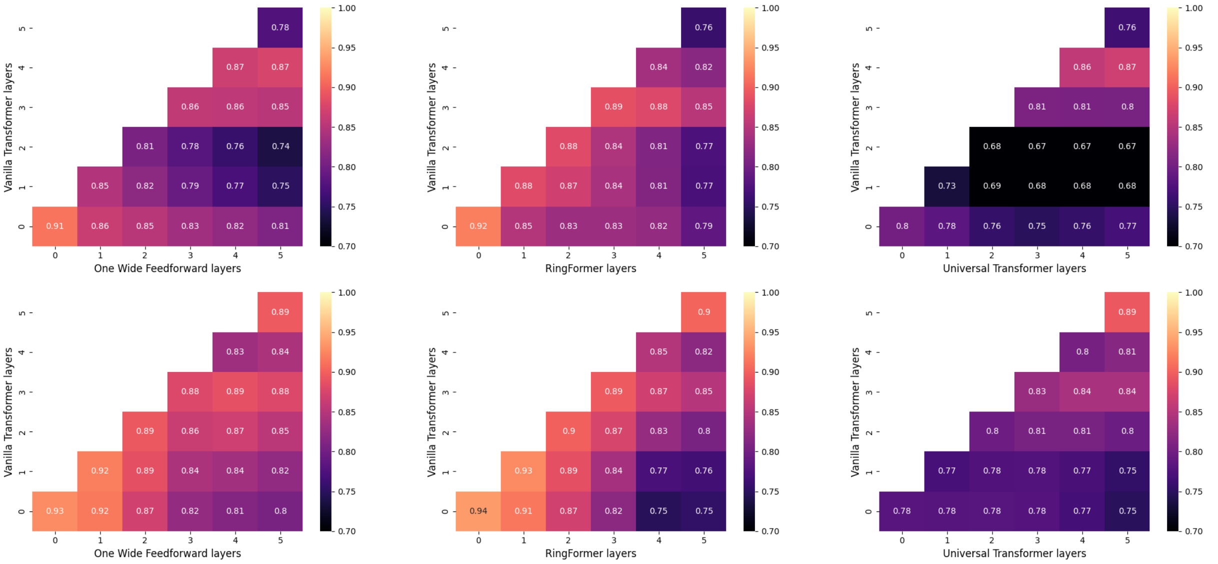

Figure 2: Representation Similarity Analysis using CKA (centered kernel alignment) (Kornblith et al., 2019) for the base-size models trained on the translation task. The figures on the upper row are for the encoder part. The figures on the lower row means are for the decoder part. All models have 6 number of layers / iterations. The values on the figures are between 0 and 1, where higher values indicate more similarity of layers between models.

$model$	H/FF	P^*	$BLEU \uparrow$
Vanilla Transformer	512 / 2048	44.05M	**30.46**
One Wide FFN	512 / 2048	20.98M	29.54
Universal	512 / 2048	7.34M	29.12
RingFormer	512 / 2048	8.94M	29.52
Vanilla Transformer	1024 / 4096	176.18M	**30.96**
One Wide FFN	1024 / 4096	83.91M	29.88
Universal	1024 / 4096	29.37M	29.47
RingFormer	1024 / 4096	35.71M	29.96

Table 1: Translation results on WMT-14 De-En (Bojar et al., 2014). We evaluated models based on test dataset BLEU score (Papineni et al., 2002), which is rounded to the second decimal place. **Bolded** score indicates the highest performance, underlined score indicates the second highest performance. The H, FF, and P^* represent the hidden input dimension, feedforward block dimension, parameter size (except parameters of embedding layer in encoder, decoder and vocabulary head), respectively.

$model$	H/FF	P	$Acc \uparrow$
ViT	512 / 2048	19.36M	**63.66%**
UiT	512 / 2048	3.60M	58.64%
OWF^d	376 / 1024	4.51M	58.62%
RingFormer	512 / 2048	4.4M	60.66%
ViT^d	328 / 1536	8.94M	62.22%
UiT^s	848 / 3072	8.84M	59.38%
OWF	512 / 2048	8.86M	61.50%
RingFormer^s	728 / 3072	8.82M	**62.58%**

Table 2: Image classification results on *ImageNet-small* (the subset of ImageNet-1K (Deng et al., 2009)). **Bolded** score indicates the highest performance, underlined score indicates the second highest performance. The superscripts "d" and "s" represent that the models are downscaled and upscaled, respectively. The H, FF, and P represent the hidden input dimension, feedforward block dimension, and total parameter size, respectively. The values for P and Acc were rounded to the second decimal place.

model (Dehghani et al., 2019), while having similar parameter size. These results also imply that our design choice for level-signals is more effective than adding input-independent sinusoidal vectors.

Image Classification The experimental results for image classification are shown in Table 2 and 3.

Using *ImageNet-small*, we conducted experiments on the ViT (Dosovitskiy et al., 2020) model, downscaled One Wide FFN (OWF^d) (Pires et al., 2023), UiT (Dehghani et al., 2019) and our RingFormer model. The results, presented in the upper half of Table 2, indicate that the ViT model achieves the highest accuracy, which is expected as it has more than four times the number of parameters compared to the other models. However, our RingFormer model has the second best performance, outperforming the other models of the same size. In the below half of Table 2, where we scale all the models to the size of One Wide FFN model, our model shows the best performance, which shows the effectiveness of our approach.

We observed similar tendency when we trained

model	H/FF	P	Acc ↑
ViT	768 / 3072	86.42M	**65.65%**
OWF	768 / 3072	34.45M	<u>64.31%</u>
UiT	768 / 3072	8.45M	61.63%
RingFormer	768 / 3072	12.02M	63.68%
UiT^s	1560 / 6240	31.99M	63.30%
RingFormer^s	1284 / 5120	31.95M	**65.91%**

Table 3: Image classification results on *ImageNet-1K* (Deng et al., 2009)). **Bolded** score indicates the highest performance, underlined score indicates the second highest performance. The superscript "s" represent that the models are upscaled. The H, FF, and P represent the hidden input dimension, feedforward block dimension, total parameter size, respectively. The values for P and Acc were rounded to the second decimal place.

bigger size models on the *ImageNet-1K* dataset, for which the results are shown in Table 3. When comparing the models with the same input hidden dimension and feedforward block dimension, ViT model showed the best result, but when we upscaled our RingFormer model (RingFormer^s) to match the size of OWF model, it outperformed the two baseline models (OWF and UiT^s), and also showed slightly higher performance compared to the ViT model.

FLOP Comparison We calculated forward GFLOPs for ViT, OWF, UiT, and RingFormer (all with the same H/FF as in Table 3) using 224×224 RGB images with 16×16 patches. ViT, UiT, and OWF have similar costs (17.64 GFLOPs), while RingFormer requires slightly more at 19.03 GFLOPs due to its depth- and input-dependent level signals. However, since these signals are low-rank, their impact on latency is minimal and becomes negligible as model size increases, with recurrent layers dominating computation.

Representation Similarity Analysis To analyze representations across layers / iterations between the original Transformer model and other models, we utilized CKA (Kornblith et al., 2019) method as shown in Figure 2. We performed this analysis on base size models, for which we used 3K test source-target pair of sentences from WMT-14 De-En (Bojar et al., 2014). The similarity scores on the diagonal axis in the sub-figures indicate how close the layers (sharing the same index) are between models. We found that RingFormer closely matches the Vanilla Transformer (Vaswani et al., 2017) along with One Wide FFN (Pires et al., 2023), while Universal Transformer (Dehghani et al., 2019) shows lower similarity. We also report the analysis results for large models in Appendix A.2.

Mean Attention Distance Analysis To study the qualities of attention heads in the vision models, we perform MAD analysis, which is conducted in the original ViT paper (Dosovitskiy et al., 2020). We first do the analysis on the smaller models trained on *ImageNet-small* (ViT, UiT^s, OWF, and RingFormer^s shown in Table 2), and also on the larger size models trained on *ImageNet-1K* (ViT, OWF, UiT^s and RingFormer^s in Table 3). We computed mean attention distances of 500 images randomly taken from the *ImageNet-small* validation set and took their average. The MAD analysis plots for each model above are shown in Figure 3.

We observe that, in the ViT, attention heads show varying attention distances, indicating use of both local and global image information. Deeper layers shift toward more global focus—a pattern also seen in One Wide FFN, as its attention layers are non-recurrent. RingFormer shows similar behavior to ViT, supporting our hypothesis that level signals effectively guide recurrent models across iterations. In contrast, Universal Transformer's signals fail to replicate this behavior, resulting in attention patterns that differ notably from the ViT model.

4.3 Ablation Study

We perform an ablation study on the translation task to assess our method's effectiveness. Training follows the main setup but uses reduced hidden and feedforward dimensions. Model settings and results are shown in Table 4.

First, we train a recurrent Transformer using static level signals introduced in Universal Transformers (Dehghani et al., 2019), which has the lowest performance. When we drop either attention level signals or FFN level signals, "w.o. attn" and "w.o. FF" in Table 4, the performance degradation occurs compared with other variations where those signals are present. Also, we do the following two ablations: 1) we add level signals "before attn" projection while keeping our original design for FF level signals, 2) we add level signals, "inter-FF signal", after intermediate feedforward projection like $FFN(x) = \sigma(xW_{up} + g_{Fi}(x))W_{down}$, while keeping our original design for attention level signals. The performances of those two experiments are almost the same but lower than our design choice, where additions occur i) after attention projection and ii) before the up-projection layer of the

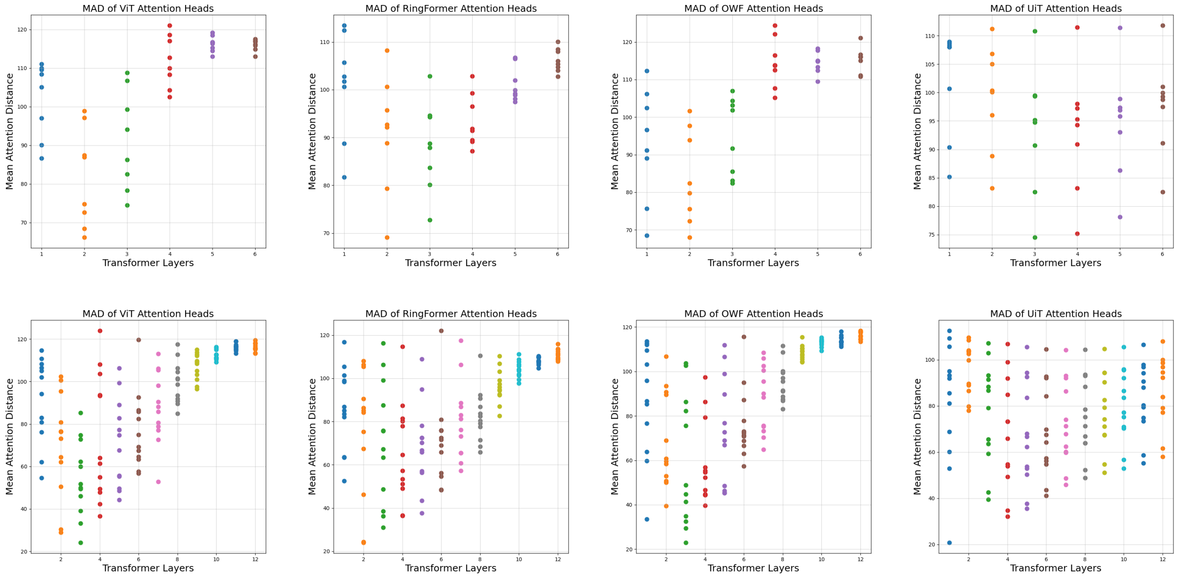

Figure 3: MAD (mean attention distance) analysis for the models trained on image classification task: ViT, RingFormer, OWF - One Wide FFN, UiT - Universal Transformer. The smaller models shown on the upper row have 8 attention heads, and larger models shown on the lower row have 12 attention heads. The points on the plots represent the mean attention distance of an attention head belonging to a particular Transformer layer.

FF block. When we use a smaller rank, H / 32, compared to our default rank, H / 16, the performance decreases, but when we increase the rank or make the matrix full-rank to generate level signals, as expected, the models show better performance.

Notably, our RingFormer model with a default H / 16 rank signals uses less than half the parameters of the full-rank version and achieves comparable performance. While the absolute gap in parameter count between full-rank and low-rank signals is modest for the small hidden dimension in Table 4, it becomes considerably bigger when scaling up the input hidden dimension. Thus, we assume that a good trade-off between performance and total model size can be achieved by using low-rank decomposition. Additionally, we can see that even with twice less core model parameter size, the performance of the full-rank RingFormer model was very close to that of the vanilla Transformer. Therefore, we expect that the capacity of our RingFormer model can be easily scaled with higher ranks while maintaining efficiency and showing similar performance relative to the original Transformer.

model	H/F	P^*	$BLEU\uparrow$
static signal	128 / 512	0.46M	23.35
w.o. attn signal	128 / 512	0.49M	24.23
w.o. FF signal	128 / 512	0.57M	24.37
before attn	128 / 512	0.56M	24.56
inter-FF signal	128 / 512	0.54M	24.58
H / 32 rank signal	128 / 512	0.51M	24.21
H / 16 rank signal	128 / 512	0.56M	24.92
H / 8 rank signal	128 / 512	0.66M	24.96
full-rank signal	128 / 512	1.25M	25.37
Vanilla Transformer	128 / 512	2.75M	25.48

Table 4: Ablation experiment results of translation task in WMT-14 (Bojar et al., 2014) German-English pairs with various model-designs. Each model is evaluated by BLEU (Papineni et al., 2002) score on the test set. The H, FF, and P^* represent the hidden input dimension, feedforward block dimension, parameter size (except parameters of embedding layer in encoder, decoder and vocabulary head), respectively.

5 Conclusion

In this paper, we introduce *RingFormer*, a parameter-efficient recurrent Transformer architecture that employs a single Transformer layer recurrently while integrating input-dependent signal vectors created using low-rank matrices for each level. This approach significantly reduces the number of parameters while maintaining high performance in tasks such as machine translation and image classification. We hope that our research on enhancing recurrent Transformer with adaptive level signals can enable smaller organizations and research institutions to train powerful models without the need for extensive computational resources, thus democratizing access to advanced AI capabilities.

6 Limitations

Our approach introduces additional computations compared to the original Transformer due to the integration of depth-specific and input-dependent signals. However, the bulk of the compute cost will be in the recurrent attention and feedforward projections, and generating low-rank signals remains efficient since the level-signal rank is much smaller than the input hidden dimension. We also note that parameter reduction can reduce memory bandwidth overhead, which is often a key bottleneck in large-scale training or edge deployments, making RingFormer efficient in many practical scenarios while maintaining comparable performance with the original Transformer.

Due to computational constraints, we were not able to conduct experiments on large-scale language modeling tasks, which require significantly more data and training resources. While our design choices and positive results across different model scales in two domains, machine translation and image classification, with supporting extensive analysis, suggest that RingFormer should retain its advantages at larger scales, future work can focus on further validating its performance on billion-parameter models and explore its effectiveness in domains such as language modeling.

References

Anthropic. 2023. Model card and evaluations for claude models. https://www-cdn.anthropic.com/5c49cc247484cecf107c699baf29250302e5da70/claude-2-model-card.pdf.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. 2021. Decision transformer: Reinforcement learning via sequence modeling. *arXiv preprint arXiv:2106.01345*.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. 2023. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, pages 7480–7512. PMLR.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. 2019. Universal transformers. *Preprint*, arXiv:1807.03819.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. *Preprint*, arXiv:2012.14913.

Gemini Team Google. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *Preprint*, arXiv:2403.05530.

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A Roberts. 2024. The unreasonable ineffectiveness of the deeper layers. *arXiv preprint arXiv:2403.17887*.

Dan Hendrycks and Kevin Gimpel. 2023. Gaussian error linear units (gelus). *Preprint*, arXiv:1606.08415.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. An empirical analysis of compute-optimal large language model training. *Advances in Neural Information Processing Systems*, 35:30016–30030.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Yifan Jiang, Shiyu Chang, and Zhangyang Wang. 2021. Transgan: Two transformers can make one strong gan. *arXiv preprint arXiv:2102.07074*, 1(3).

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *Preprint*, arXiv:1412.6980.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. 2019. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMLR.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, and et al. 2024. Gpt-4 technical report.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. *Preprint*, arXiv:1806.00187.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

William Peebles and Saining Xie. 2022. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*.

Telmo Pessoa Pires, António V. Lopes, Yannick Assogba, and Hendra Setiawan. 2023. One wide feedforward is all you need. *Preprint*, arXiv:2309.01826.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Machel Reid, Edison Marrese-Taylor, and Yutaka Matsuo. 2021. Subformer: Exploring weight sharing for parameter efficiency in generative transformers. *Preprint*, arXiv:2101.00234.

Sho Takase and Shun Kiyono. 2021. Lessons on parameter sharing across layers in transformers. *arXiv preprint arXiv:2104.06022*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Haoran Xu, Benjamin Van Durme, and Kenton Murray. 2021. Bert, mbert, or bibert? a study on contextualized embeddings for neural machine translation. *Preprint*, arXiv:2109.04588.

Bowen Zhang, Shuyang Gu, Bo Zhang, Jianmin Bao, Dong Chen, Fang Wen, Yong Wang, and Baining Guo. 2022. Styleswin: Transformer-based gan for high-resolution image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11304–11314.

A Appendix

A.1 Implementation Details

Translation Models are trained based on the two size variations, base size and large size. The base size models are trained based on the following model configuration settings: 6 Transformer layers, 8 attention heads, 512 hidden dimension size, 2048 feedforward dimension with maximum sequence length 50. For training, their maximum learning rate is 7e-4 with 17K step cosine warm-up scheduler and total 210K training steps on two A100 80GB GPUs. The large size models are trained based on the following model configuration settings: 6 Transformer layers, 16 attention heads, 1024 hidden dimension size, 4096 feedforward dimension with maximum sequence length 50. For training, their maximum learning rate is 2e-4 with 17K step cosine warm-up scheduler and total 210K training steps on two A100 80GB GPUs.

Image Classification For the models trained on *ImageNet-small* dataset, we used 224x224 image resolution, 16x16 patch size, 6 Transformer layers, 8 attention heads, learning rate of $1e^{-3}$, cosine learning rate scheduler with 2K warm-up steps, batch size of 1024, and training for 9775 steps (100 epochs) with one RTX 3090 GPU. For the models trained on *ImageNet-1K* dataset, we used the same image resolution and patch size as mentioned above, 12 Transformer layers, 12 attention heads, learning rate of $5e^{-4}$, cosine learning rate scheduler with 3128 warm-up steps (5 epochs), batch size of 4096, 16 gradient accumulation steps, and training for around 15650 steps (50 epochs) on two RTX 3090 GPUs.

For all models, we used dropout rate of 0.1, gradient clipping of 1.0 during training and GELU (Hendrycks and Gimpel, 2023) activation function.

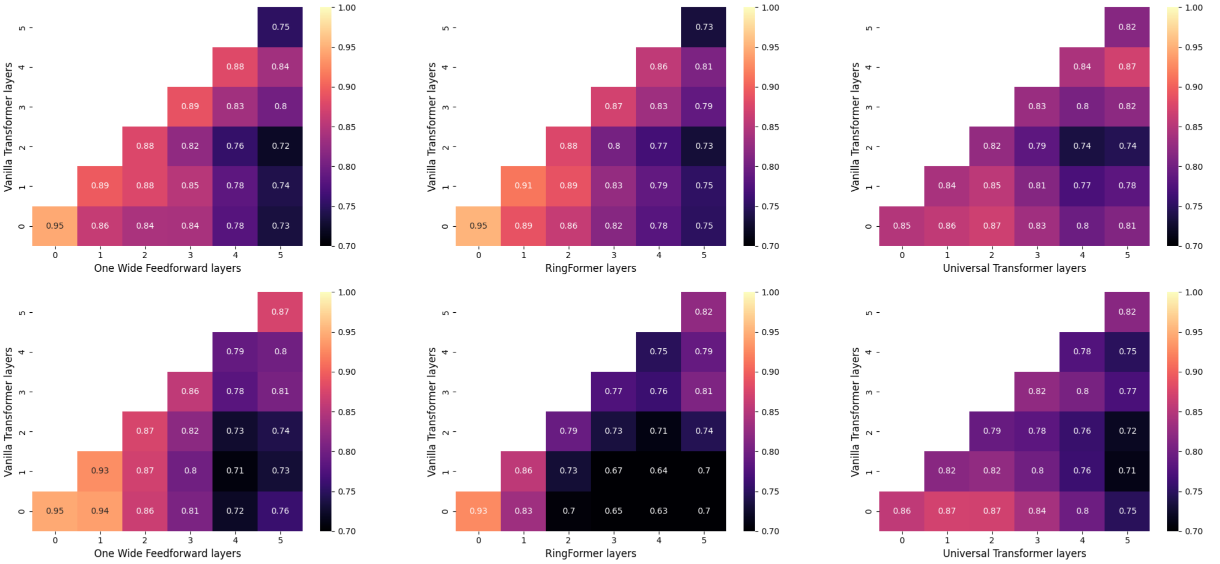

Figure 4: Representation Similarity Analysis using CKA (centered kernel alignment) (Kornblith et al., 2019) for the large-size models trained on the translation task: Transformer, Ring - RingFormer, OWF - One Wide FFN, Uni - Universal Transformer. The figures on the upper row are for the encoder part. The figures on the lower row means are for the decoder part. All models have 6 number of layers / iterations. The values on the figures are between 0 and 1, where higher values indicate more similarity of layers between models.

A.2 Additional Analysis

In Figure 4, we share the representation similarity analysis for big size models in the Translation task. This analysis also has been conducted under the same conditions as in the base size case. Similar with the results in Figure 2, One Wide FFN (Pires et al., 2023) and our RingFormer model have higher layer-wise representations with the Vanilla Transformer (Vaswani et al., 2017) compared to Universal Transformer (Dehghani et al., 2019).

A.3 ImageNet-small Dataset

We sampled a subset of *ImageNet-1K* (Deng et al., 2009) that contains randomly selected 100 classes, with 100,000 images for training and 5000 images for testing, in order to perform experiments on smaller size models. In the supplimentary **code**.zip file, we will share the names of all the sampled images for training and testing as a json file.