Relaxed Recursive Transformers

Sangmin Bae*, Adam Fisch, Hrayr Harutyunyan, Ziwei Ji, Seungyeon Kim, Tal Schuster†

*Work done during an internship at Google DeepMind, †Corresponding Author

Paper

Adaptive Computation: Early-exiting Framework

- To address the computational demands of large language models (LLMs), various adaptive computation methods are being proposed to enhance efficiency.
- **Early-exiting** framework:
  - **Easy** tokens exit in **early layers**, while **hard** tokens forward **whole depths**.

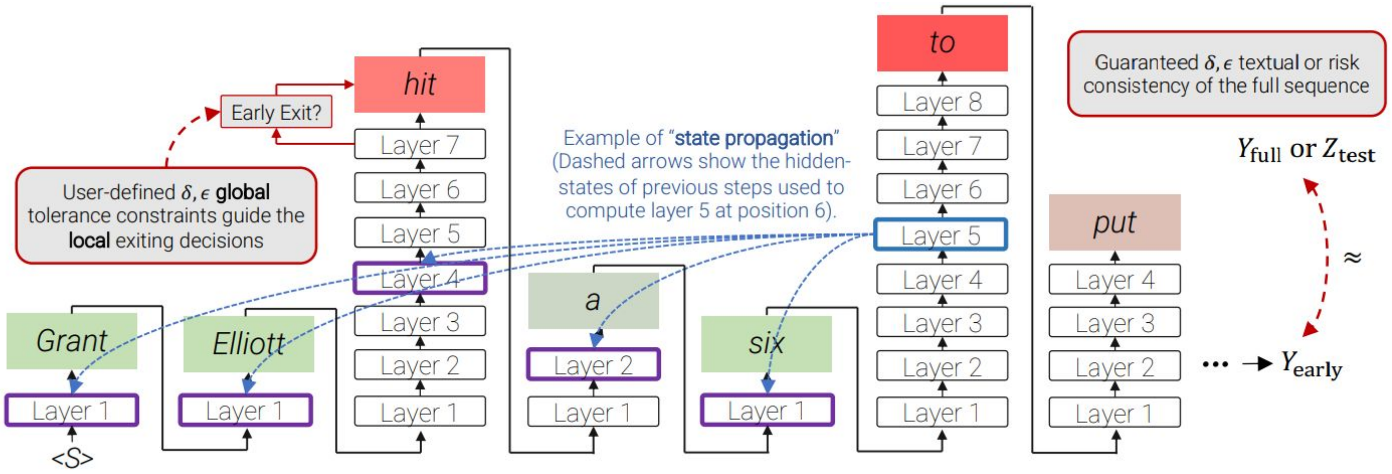
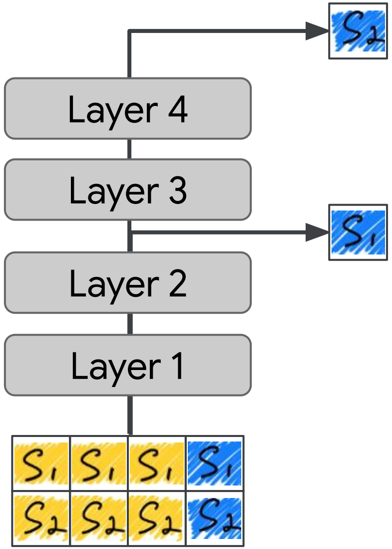


Figure from "Schuster, Tal, et al. Confident Adaptive Language Modeling (NeurIPS 2022)".

Batched Inference is Tricky for Early-exiting

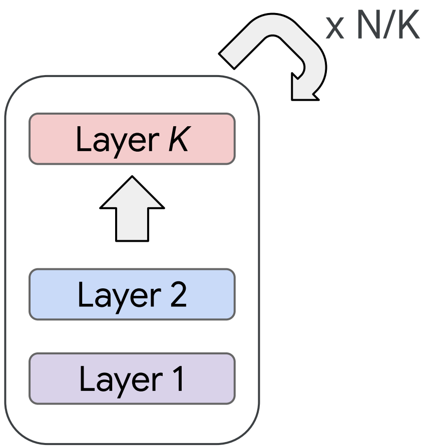
- **Dynamic exit points** paradoxically poses a **challenge for batched inference**.
  - Although the tokens exit at early layers, they **must wait for the others**.



⏳S_1 waits until S_2 is finished,
before proceeding to the next sequence.

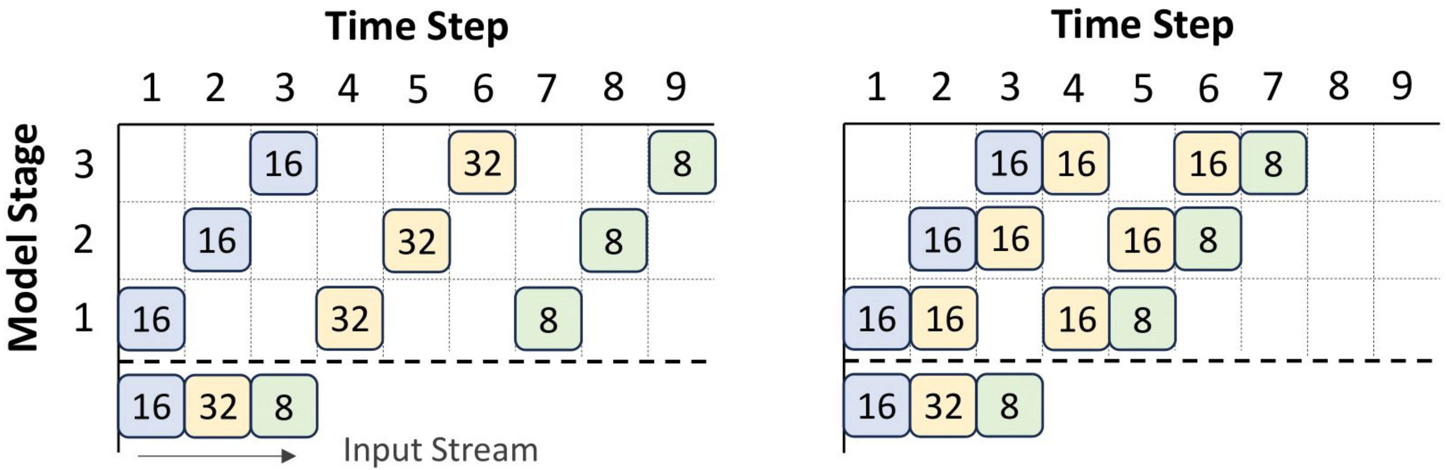
Solution: Parameter Sharing

- **Recursive Transformers:**
  - Recursively apply the **same** function for ***N/K*** times.
  - Reduce parameter size by *N/K*, **minimizing memory footprint**.



Recursive Transformers Enable Continuous Depth-wise Batching

- Since the model's depths (stages) share the same parameters, we can **maximize throughput** by **continuously batching samples at different depths**.



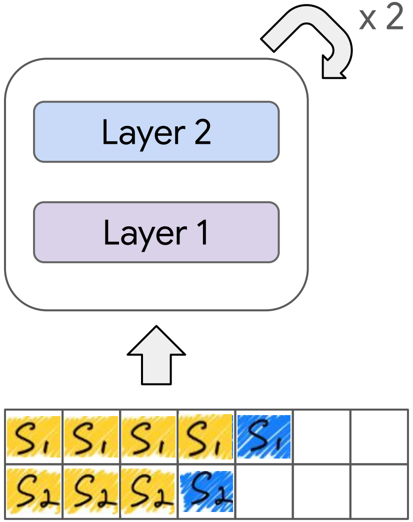
(a) Vanilla Batching

(b) Depth-wise Batching

Early-exiting Further Enhances Continuous Depth-wise Batching

S_1 early-exits after first iteration.

S_2 have to forward twice for an accurate prediction.



S_1 **of 5th sequence** and S_2 **of 4th sequence** will be batched together!

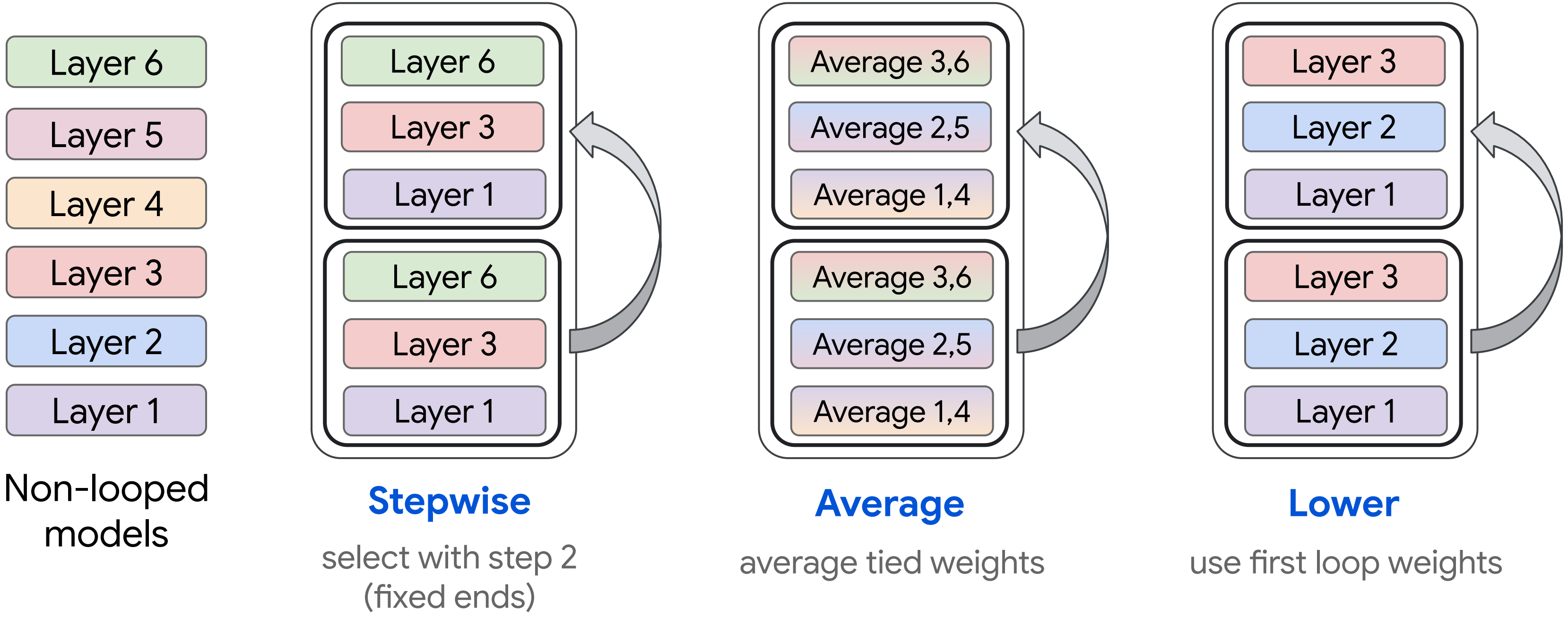
Two Research Goals for Recursive Transformers

- **Performance**
  - We need to find a **correct recipe to train** Recursive Transformers.
  - This includes **initialization** methods and **relaxation** strategies for weight tying.

- **Throughput**
  - By pairing an **early-exit** framework with **continuous batching**, we aim to improve inference speed.

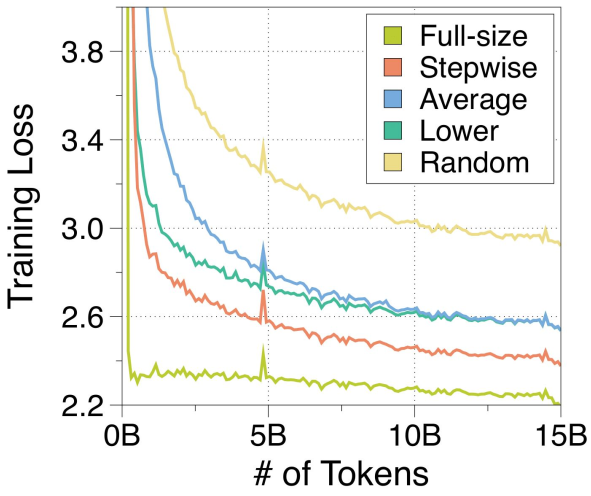
Initialization Techniques for Looped Layers

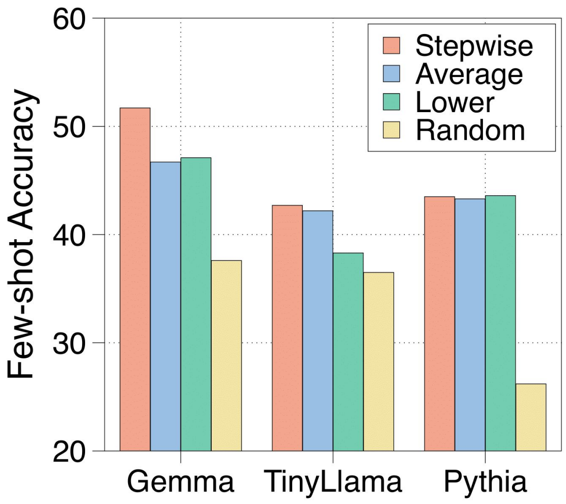
- We **convert existing LLMs** into Recursive Transformers using following initialization.



Takeaways for Recursive Transformers

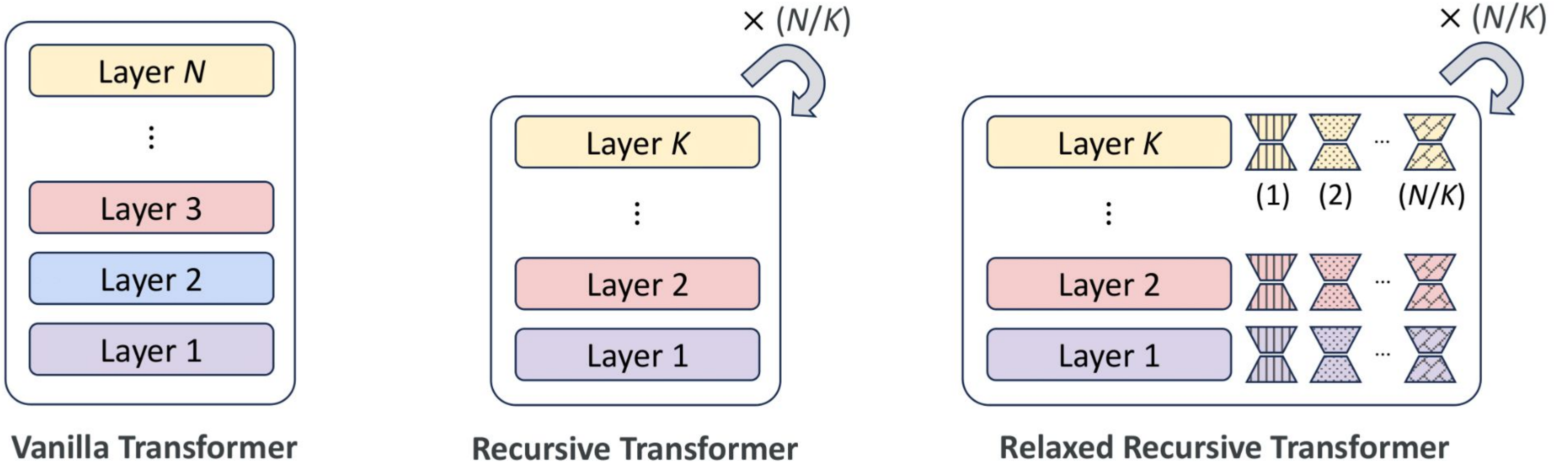
We find that **converting well-pretrained models** into Recursive Transformers leads to **high-performing models with minimal uptraining**. Notably, Initializing looped layers via the **Stepwise method** yields the best results.





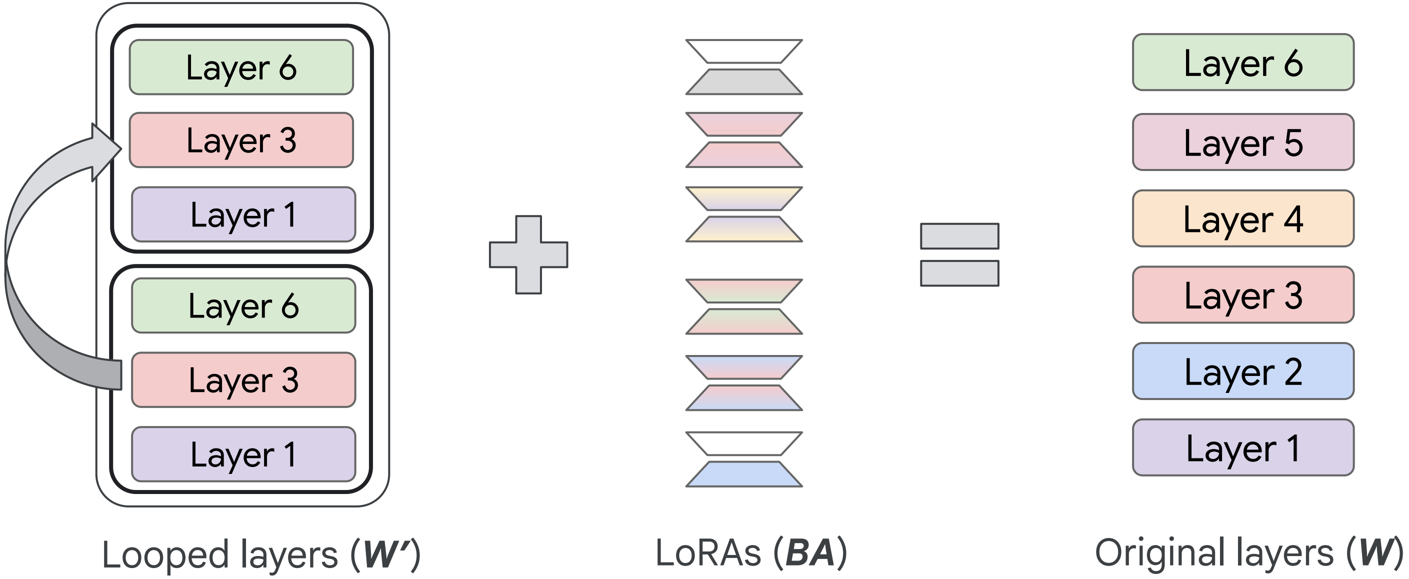
Relaxed Recursive Transformers with Layer-wise LoRAs

- We augment perfectly tied layers with **specific LoRAs corresponding to each loop**.



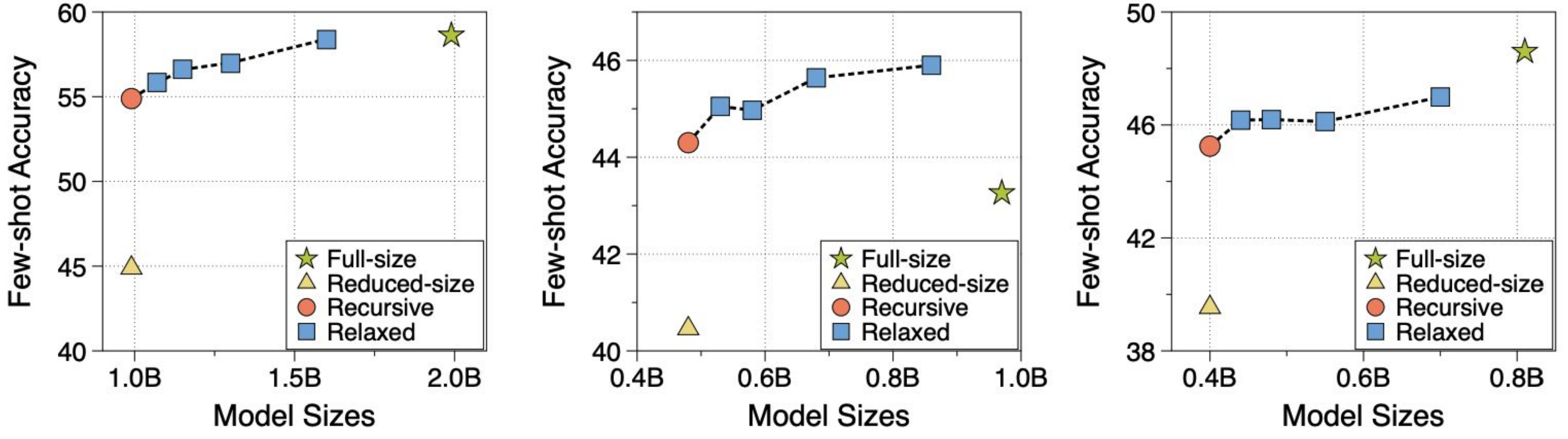
Truncated SVD to Initialize LoRA Modules

- **Truncated SVD** on residual matrices (***W* - *W′***) matrices → (***UΣ*** for ***B*** and ***V*** for ***A***).



Takeaways for Relaxed Recursive Transformers

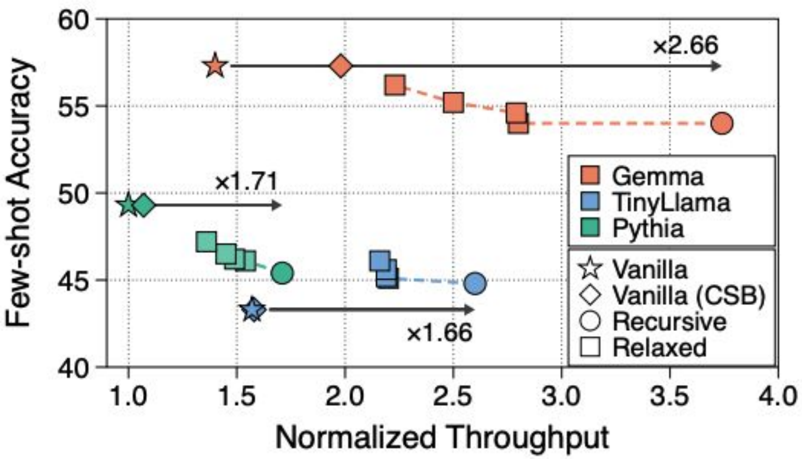
Our **SVD-based initialization** allows for **smooth transition between vanilla and recursive models** by **adjusting LoRA ranks**. Initializing looped layers with **Average method** leads to the best performance in this relaxed setting.



(a) Gemma (b) TinyLlama (c) Pythia

Takeaways for Continuous Depth-wise Batching

In theory, we can achieve up to **2-3x speedup** compared to a vanilla Transformer via **continuous depth-wise batching** and **early-exiting**.



N-emb	Loop	LoRA	Batch	Exit	Acc.	Thr.	Δ_V	Δ_{Seq}
1.99B	-	-	-	✗	57.3	1080	×1.00	×0.71
1.99B	-	-	CSB	✗	57.3	1528	×1.41	×1.00
0.99B	2	-	CDB	✓	54.0	2877	×2.66	×1.88
1.07B	2	64	CDB	✓	54.0	2157	×2.00	×1.41
1.15B	2	128	CDB	✓	54.6	2149	×1.99	×1.41
1.30B	2	256	CDB	✓	55.2	1921	×1.78	×1.26
1.60B	2	512	CDB	✓	56.2	1719	×1.59	×1.13

Conclusion

- We introduce Recursive Transformers, in which we compress LLMs via parameter sharing across recursively looped blocks of layers.
- We present a novel relaxation strategy that allows for low-rank deltas between shared layers by integrating layer-specific LoRA modules into the fully-tied structure.
- By exploiting the recursive patterns and an early-exiting approach, we propose a continuous depth-wise batching paradigm tailored for efficient serving systems of Recursive Transformers.

Relaxed Recursive Transformers

Sangmin Bae*, Adam Fisch, Hrayr Harutyunyan, Ziwei Ji, Seungyeon Kim, Tal Schuster

Poster session is scheduled for April 26th (Saturday) 10 am – 11:30 am (SGT)

Paper